Apple-Works Forum

The Monthly Publication of **NAUG:** ***The National AppleWorks Users Group***

Volume VI, No. 1 **Four Dollars**

TABLE OF CONTENTS

Support for AppleWorks and ///EZ Pieces Users

How to Reduce Disk Swapping

Dear Cathleen,

Here is a suggestion that can reduce disk swapping on Apple II systems equipped with 512K or less of memory, one 3.5-inch disk drive, and one 5.25-inch disk drive.

AppleWorks will sometimes try to access its Program Disk when you load large data base files onto the desktop from a 3.5-inch data disk. You must then remove the data disk from the drive, let AppleWorks read a segment from the Program Disk, and then reinsert the data disk to complete the loading operation.

Here's a trick that can eliminate the disk swapping.

Copy the files SEG.DB and SEG.AW onto a 5.25-inch disk that has the same name as the disk you used to load AppleWorks (usually /APPLEWORKS). Leave that disk in the 5.25-inch disk drive. AppleWorks will find the files it needs during the load process and you will not have to swap disks.

Jack Olson
St. Louis, Missouri

[Ed: If you have 512K or more of memory in your computer, you should configure AppleWorks so all the program modules load into memory upon bootup. If AppleWorks accesses your Program Disk after it loads the modules, your computer does not have enough memory to store both the program and your data; you should consider getting additional memory.

Remember that you can also use your 5.25-inch disk drive to store your data files. That lets you leave the AppleWorks Program Disk in the 3.5-inch drive. It also eliminates the need for disk swapping when you spell check a document and when you work with large files.]

The ***National AppleWorks Users Group*** **(NAUG)** is an association that supports AppleWorks users. **NAUG** provides technical support and information about AppleWorks and enhancements to that program. Our primary means of communicating with members is through the monthly newsletter entitled the ***AppleWorks Forum***.

Printer Doesn't Page Properly

Dear NAUG:

My ImageWriter II gives me problems printing word processor documents. The first page always prints correctly, then the printer scrolls to the middle of the second page and continues printing. This throws off the alignment of the second and all following pages. The problem does not occur with multiple-page data base or spreadsheet files.

Do you have any idea what causes this problem?

John Nied
Danville, Pennsylvania

[Ed: This problem is usually caused by incorrect interface card or DIP switch settings. Try the following:

1. *Configure AppleWorks so the "Accepts top of page command" on the Printer Setting Menu is set to "No".*
2. *Turn off the "Perforation skip" setting on your printer. Switch 1-5 controls perforation skip on ImageWriter II printers; set that switch to "Open".*

The DIP switches in the ImageWriter II should be set as follows:

	1	2	3	4	5	6	7	8
Switch 1:	Up	Up	Up	Up	Up	Down	Up	Up
Switch 2:	Down	Down	Up	Up	Down	Up		

Hopefully these settings will solve your printing problems.]

AppleWorks Forum

Editor: Cathleen Merritt
Associate Editor: Warren Williams
Page Layout: Nanette Luoma
Publisher: The National AppleWorks Users Group

The "AppleWorks Forum" (ISSN 0893-4118) is published monthly
for $29 per year by the National AppleWorks Users Group,
49068 Harvest Dr., Plymouth, MI 48170.
Second Class postage paid at Plymouth, MI, and additional mailing offices.
POSTMASTER: Send address changes to AppleWorks Forum,
NAUG, Box 87453, Canton, MI 48187

Spreadsheet Tips

How to Use @PMT and @TERM

by Stan Hecker

This is the third in a series of articles that describe how to use the new financial functions in AppleWorks 3.0. The author assumes that you read the previous articles in this series.

Last month, I described how to use AppleWorks' @FV function to determine the future value of a series of known payments. I assumed that you knew how much you could save, the interest rate you would earn, and how long you planned to save.

This month I will describe the @PMT and @TERM functions; two functions that complement the power of @FV. If you know your financial goal and the interest rate, @PMT and @TERM can determine how much and how long you must save to reach that goal. I will also use these functions to help describe "present value" annuities such as bank loans.

Syntax of @PMT

@PMT determines the amount you must save each period to accumulate a specified sum of money or the amount you must pay to repay a loan. The syntax of @PMT statements is:

```
@PMT(rate, term, present value [, future value, type])
```

where "rate" is the interest rate you expect to earn each period, "term" is the number of payment periods, and "present value" is the current value of the account. "Future value" and "type" are optional. "Future value" is the amount of money you want to have at a specific time in the future. "Type" is "Ø" or blank if you will save the money at the end of the each period and "1" if you will save the money at the beginning of each period.

An Example of @PMT

Let's use @PMT to determine how much money you must save to end up with $4,000 for a family vacation you want to take a year and a half from now. Imagine that you already have $1,652 in a 7% money market account which compounds monthly. How much do you need to put into the account each month to end up with $4,000 at the end of 18 months?

Put the cursor on any spreadsheet cell and enter the formula:

```
@PMT(.Ø7/12,1.5*12,-1652,4ØØØ,1)
```

The "rate" in this formula is 7% divided by 12 months; written as .07/12.

The term is 1.5 years, or 18 months. You can enter "18" directly into the spreadsheet or you can type "1.5*12". That keeps the interest rate and the term of the transactions in the same time period.

The present value is $1,652 and is entered as -1652 because it is money out of your pocket.

@PMT and @TERM can determine how much you should save or borrow.

The future value is $4,000, a positive value because it will be money in your pocket when vacation time arrives.

The type designation is 1 because you will make your deposits at the beginning of each month.

When you press the Return key, AppleWorks returns a value of -113.80. Under the conditions stated, you will have to save $113.80 each month to accumulate the $4,000 required for the vacation.

If You Have No Savings

Now imagine that you don't have any money to start your account. How much must you save each month to accumulate the $4,000?

If you guessed that the present value figure changes from -1652 to zero, you are correct. Here is the new formula:

```
@PMT(.Ø7/12,1.5*12,Ø,4ØØØ,1)
```

Note that present value is a required element in this financial function; you must enter a present value, even if it is zero.

When you press the Return Key, AppleWorks will display -210.18, a new, larger amount that you must save each month to pay for a $4,000 vacation a year and a half from now.

@TERM Syntax

The @TERM function computes (a) how long a stream of level payments must continue to reach your savings goal, or (b) how long it will take to repay a loan. The full syntax of @TERM is:

```
@TERM(rate, payment, present value [, future value, type])
```

where "rate" is the expected interest rate per period, "payment" is the amount you will save each period, "present value" is the current amount saved, "future value" is the total amount you want to save, and "type" indicates whether you will put the money aside at the beginning or end of each period. Once again, the "future value" and "type" elements are optional; I will describe formulas that do not use those elements later in this article.

> *"AppleWorks can answer your loan and savings questions in seconds."*

An Example of @TERM

Here is an example of a situation that requires @TERM: Imagine that your teenager is an industrious worker who mows lawns and babysits to earn extra money. Your teen is musically inclined, wants to buy a MIDI music interface for your computer, and can save $22 each week toward the $650 purchase. Also assume that passbook savings accounts pay 6.5%, compounded weekly. How long will your teenager have to save $22 per week to buy the hardware?

As a rough estimate, your pocket calculator tells you that $650 divided by $22 results in just under 30 payments without considering interest. But will the earned interest significantly reduce the time it takes to save the $650?

Let's use the @TERM function to find out. The interest rate is 6.5% annually, compounded weekly. You express this value as a decimal fraction divided by the period: type it as .065/52. That will compute the amount of interest earned each week.

The payment is $22 per week, which is in the same time period as the interest rate. Express the payment as -22 in the @TERM formula; it is money out of your pocket.

The teenager has not yet opened an account, so the present value is zero. The future value is the teen's goal; $650. Enter the "type" element as a "1" because the deposits will be made at the beginning of each week.

The expression in the spreadsheet should look like this:

```
@TERM(.Ø65/52,-22,Ø,65Ø,1)
```

When you press the Return Key, AppleWorks will display "28.995". The weekly interest saves a little less than a week. Your teenager will accumulate the $650 savings in approximately 29 weeks.

Present Value Transactions

The transactions we considered so far are "future value transactions"; transactions where you put money away now for the future. Savings transactions are examples of future value transactions.

Next we turn to "present value transactions". In a present value transaction you get your money now and repay the money over a period of time. Loans are excellent examples of present value transactions. These transactions consist of a starting balance and a steady stream of payments that continue until there is no balance due at the end of the transaction.

The @PMT and @TERM functions can both serve as present value functions: Unless you tell AppleWorks otherwise, both functions assume that the "future value" is zero. Since most bank loans end up paid and thus have a future value of zero, you omit the future value option to solve bank loan and other present value problems.

The "type" element in both functions is also optional. When you set "type" to zero or leave "type" blank, AppleWorks assumes that the payments occur at the end of each remittance period. Setting "type" to 1, as we did in the earlier examples in this article, specified that the payments occurred at the start of each remittance period.

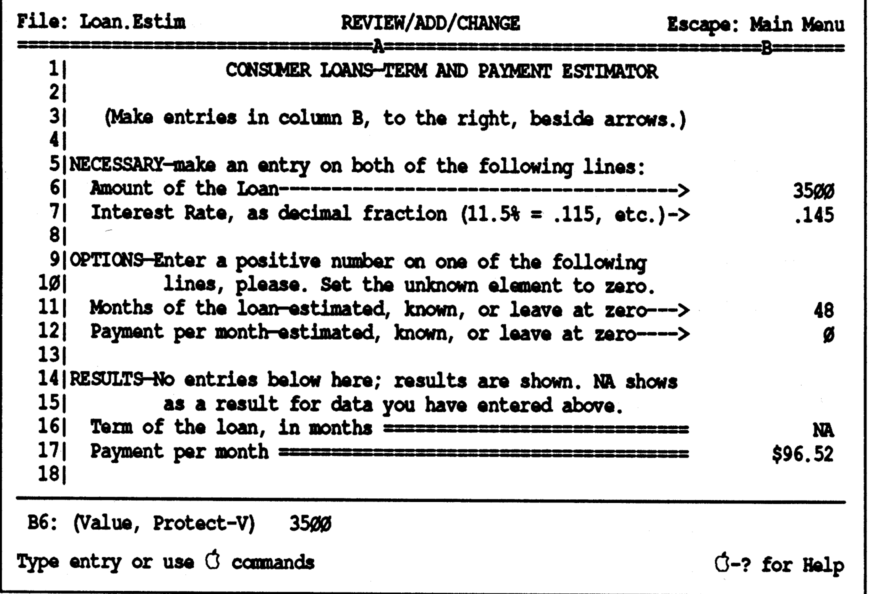
Since loan payments usually occur at the end of the month, you generally do not use the "type" element in present value calculations.

Answering Loan Questions

Consider this example that uses the @TERM function to determine how long it takes to repay a loan.

Suppose that you want to buy a used trailer for $3,500 and the bank will lend you money at 14.5% interest for up to 48 months. You can afford to repay the loan at the rate of $100 per month. Can you borrow the full amount from the bank and pay it off within 48 months?

Figure 1: A Consumer Loan Estimation Template



```
File: Loan.Estim                REVIEW/ADD/CHANGE              Escape: Main Menu
===========================A================================B======
  1|               CONSUMER LOANS—TERM AND PAYMENT ESTIMATOR
  2|
  3|   (Make entries in column B, to the right, beside arrows.)
  4|
  5|NECESSARY—make an entry on both of the following lines:
  6|  Amount of the Loan---------------------------------->          35ØØ
  7|  Interest Rate, as decimal fraction (11.5% = .115, etc.)->      .145
  8|
  9|OPTIONS—Enter a positive number on one of the following
 1Ø|          lines, please. Set the unknown element to zero.
 11|  Months of the loan—estimated, known, or leave at zero--->        48
 12|  Payment per month—estimated, known, or leave at zero---->         Ø
 13|
 14|RESULTS—No entries below here; results are shown. NA shows
 15|          as a result for data you have entered above.
 16|  Term of the loan, in months ==========================          NA
 17|  Payment per month ====================================      $96.52
 18|

 B6: (Value, Protect-V)   35ØØ

Type entry or use Ć commands                                 Ć-? for Help
```

The following formula computes the number of months you will need to pay off the loan:

```
@TERM(.145/12,-1ØØ,35ØØ)
```

The interest rate in this example is .145 divided by the number of months in a year. The payment is a negative $100 because it will be cash out of your pocket each month. The present value is positive, because it is money the bank will give you now.

If you enter this formula and press the Return Key, AppleWorks displays a value of 45.7724249. It will take almost 46 months to pay off this loan.

Computing the Loan Payment

Now let's use the @PMT function to determine what your payment will be if you take the full 48 months the bank will allow. Here is the formula that solves this problem:

```
@PMT(.145/12,48,35ØØ)
```

In this case, the interest paid is 14.5% per year divided by 12 (the number of months in a year), 48 is the number of payments, and 3500 is the amount of the loan.

When you press the Return Key, AppleWorks calculates the result as "-96.52". Your monthly payment for the 48-month loan is $96.52.

Small Differences

No matter what your results, your actual bank payment will be slightly different than the amount AppleWorks predicts. Small differences are common and result from the rounding of numbers in complex calculations, different ways of computing interest, and even the day of your call (how is part-month interest handled?). However, if the difference is larger than a few cents per month, you have good reason to request an explanation.

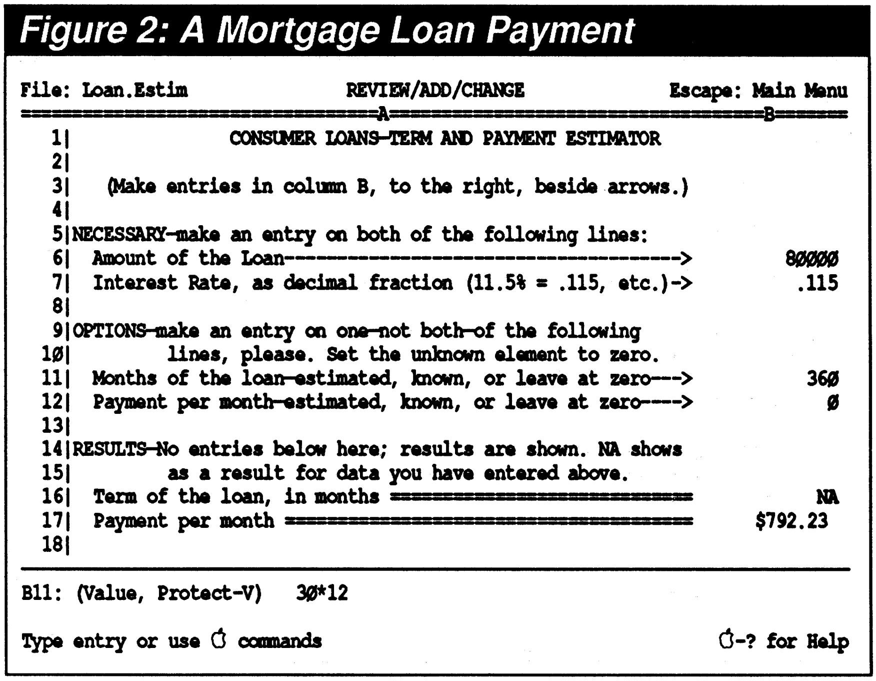
A Consumer Loan Template

Now that you understand the syntax and applications of @PMT and @TERM, let's examine two templates that use these functions.

The loan analysis template in *Figure 1* computes how long it will take to repay a loan, or the amount of your monthly payment. Unlike our earlier examples with a single formula, this template asks you to enter a number of months into cell B11 and does all the calculations necessary to compute each monthly remittance period. The template also accepts and yields "payment per month" as positive values in cells B12 and B17. These changes make the template more intuitive and easier to use.

Despite the apparent complexity of the template, there are only two formulas in *Figure 1*. Cell B16 in *Figure 1* uses the following formula to compute the term (or duration) of the loan:

Figure 2: A Mortgage Loan Payment



```
File: Loan.Estim              REVIEW/ADD/CHANGE             Escape: Main Menu
==============================A===================================B=======
  1|            CONSUMER LOANS-TERM AND PAYMENT ESTIMATOR
  2|
  3|   (Make entries in column B, to the right, beside arrows.)
  4|
  5|NECESSARY-make an entry on both of the following lines:
  6|  Amount of the Loan---------------------------------->         8ØØØØ
  7|  Interest Rate, as decimal fraction (11.5% = .115, etc.)->      .115
  8|
  9|OPTIONS-make an entry on one-not both-of the following
 1Ø|          lines, please. Set the unknown element to zero.
 11|  Months of the loan-estimated, known, or leave at zero--->        36Ø
 12|  Payment per month-estimated, known, or leave at zero---->          Ø
 13|
 14|RESULTS-No entries below here; results are shown. NA shows
 15|          as a result for data you have entered above.
 16|  Term of the loan, in months ==========================           NA
 17|  Payment per month ====================================      $792.23
 18|
---------------------------------------------------------------------------
B11: (Value, Protect-V)   3Ø*12

Type entry or use  commands                                   -? for Help
```

```
@IF(B11=Ø,@TERM(B7/12,-B12,B6),@NA)
```

The @TERM function draws the annual interest rate from cell B7 and converts that rate into a monthly figure. The formula converts the payment in cell B12 into a negative number. The value of the loan is drawn by a reference to cell B6.

The @IF function says, "If the term of the loan in cell B11 is zero, the user wants to calculate the term of the loan. In that case, use the term function. If cell B11 contains a value other than zero, the user already knows the term of the loan, so display the letters 'NA'."

The example in *Figure 1* has a known term of 48 months. Thus, cell B11 contains "48"; that activates the "false" branch of the @IF function and the letters "NA" appear in cell B16.

Figuring the Payment of a Loan

Cell B17 in *Figure 1* contains the formula:

```
@IF(B12=Ø,@ABS(@PMT(B7/12,B11,B6)),@NA)
```

The @PMT function in this formula computes the monthly interest rate by dividing the annual interest rate in cell B7 by 12. The formula draws the term of the loan from cell B11 and the amount of the loan from cell B6.

The payment of a loan is an out-of-pocket expense and is usually expressed as a negative number. However, the @ABS function in this formula tells AppleWorks to display the number as a positive value. Thus, by enclosing the @PMT statement within the @ABS function, we tell the template to display the final result as a positive number.

The @IF function says, "If the payment of the loan in cell B12 is zero, the user wants to calculate the payment. However, if cell B12 holds some value other than zero, then the user already knows the payment amount and we will display the letters 'NA'."

A Mortgage Loan

The example in *Figure 2* shows how to use the same template to estimate the principal and interest payment associated with a 30 year, $80,000 mortgage loan at 11.5%. Note that I entered "30*12" into cell B11 in this figure so AppleWorks would compute the total number of monthly payments.

I suggest that you key this template into your computer and try different parameters. For example, what is the impact if you can get a 10.5% loan instead of the 11.5% loan offered by the bank? AppleWorks answers these questions in seconds. *[Ed: NAUG on Disk users will find this template in the /AW.FORUM/TEMPLATES subdirectory.]*

Summary and a Look Ahead

This month we used AppleWorks' @PMT and @TERM functions to examine future value (savings) and present value (loan) transactions. Next month I will describe the @PV function and examine a useful template that demonstrates the combined power of the annuity functions.

[Stan Hecker is on the administrative staff at Michigan State University.]

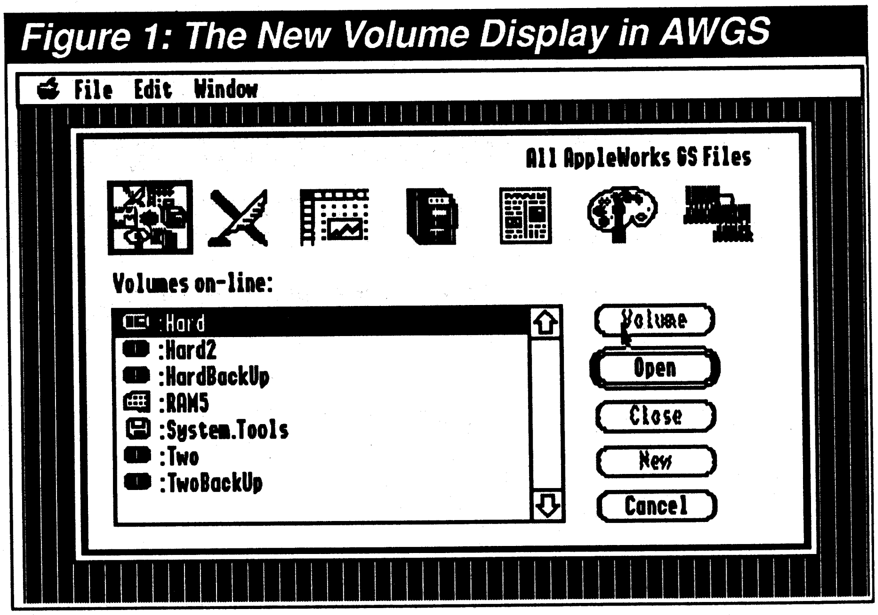
How to Install GS/OS 5.0.3

by John Link

Apple Computer recently released GS/OS 5.0.3, an update to the company's 16-bit operating system for Apple IIGS computers. This article describes some of the new features of version 5.0.3, how to install and update GS/OS on a hard drive, and how to use GS/OS with floppy disks. Members can get GS/OS 5.0.3 from the NAUG Public Domain Library ($12 plus $2 s/h).

GS/OS 5.0.3 offers some significant improvements over earlier versions of the Apple IIGS operating system. Version 5.0.3 includes numerous bug fixes and new features such as an enhanced "volume" command which automatically lists all online volumes in every sixteen bit application you open. (A click on the "Volume" button displays a list of all volumes online; you can open any volume directly from the list and no longer have to cycle through the devices to choose a volume. *Figure 1* shows the effect of this feature in AppleWorks GS.)

Figure 1: The New Volume Display in AWGS



:Hard
:Hard2
:HardBackUp
:RAM5
:System.Tools
:Two
:TwoBackUp

ImageWriter and ImageWriter LQ owners will appreciate version 5.0.3's enhanced drivers for these printers, and the SCSI drivers in version 5.0.3 are totally compatible with the popular Conner hard drive mechanisms.

The newest version of GS/OS also adds new features to the audio compression and expansion tool set (A.C.E) and improves support for AppleTalk, AppleShare, UniDisk 3.5 drives, CD-ROMs, the console display, and ExpressLoad file loading. 5.0.3 also includes version 1.9 of ProDOS and BASIC.SYSTEM 1.4.1.

One of the most significant enhancements to GS/OS 5.0.3 is not obvious to users: This is the first version of GS/OS that will run HyperCard GS.

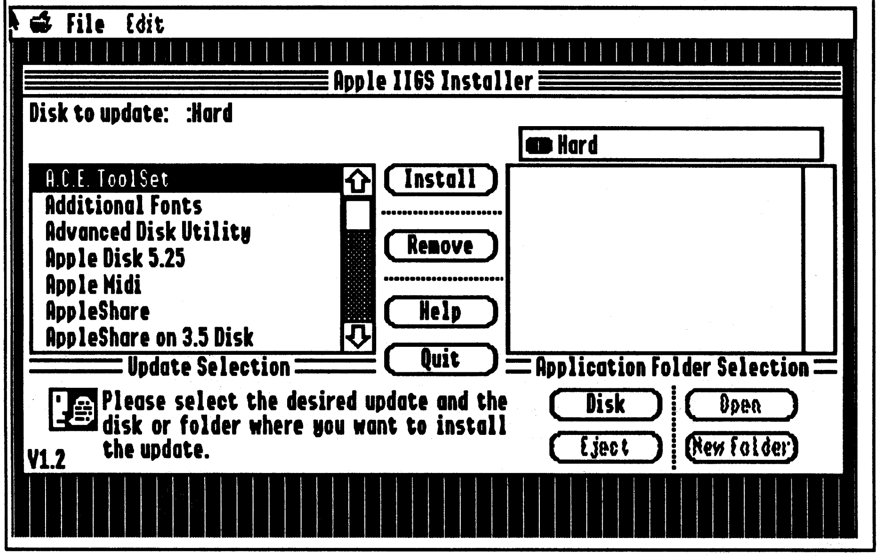
Be Careful with the Installer

In this article I will describe how to use the GS/OS Installer to install GS/OS 5.0.3 on hard disk and floppy disk drive systems. However, a word of caution before you start: The GS/OS 5.0.3 Installer does not provide adequate warning before deleting files. Both the "Install" and "Remove" buttons act after a single click and do not offer a second chance if you make a mistake. Be certain you know what you want to do before clicking on either of those buttons.

You also have to be careful about using some of the "scripts" (pre-programmed installation routines) available with the Installer. The AppleShare on 3.5 Disk, Aristotle Update, Local Network Startup, Server Network Startup, and Server Quick Logoff scripts remove some important files from the target disk and/or do an inadequate job of updating GS/OS. Make certain that you use these scripts only if you work in a networked environment that requires these GS/OS modules.

Finally, if you have only one 3.5-inch disk drive, you will have to swap disks repeatedly during the

Figure 2: GS/OS 5.0.3 Installer Screen



Apple IIGS Installer

Disk to update: :Hard

A.C.E. ToolSet
Additional Fonts
Advanced Disk Utility
Apple Disk 5.25
Apple Midi
AppleShare
AppleShare on 3.5 Disk

Update Selection

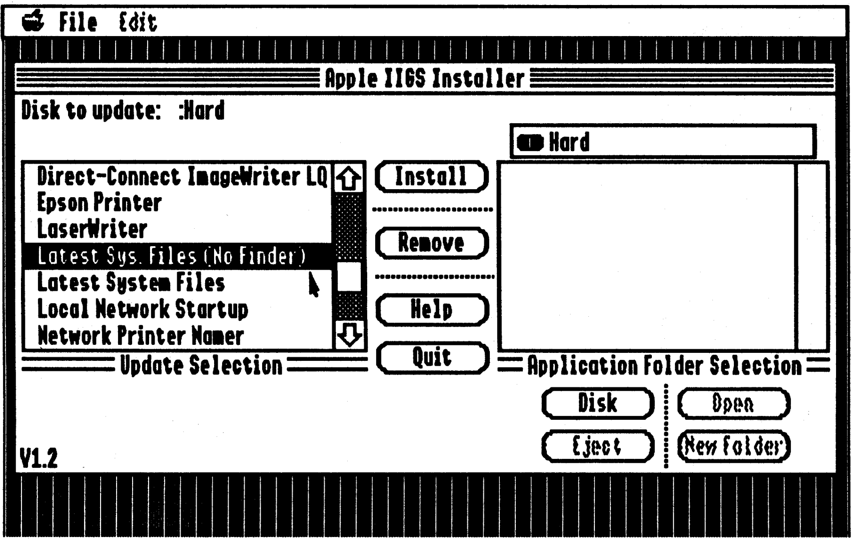
Hard

Application Folder Selection

Please select the desired update and the disk or folder where you want to install the update.

V1.2

Figure 3: Selecting the Correct Installation Script



Apple IIGS Installer

Disk to update: :Hard

Direct-Connect ImageWriter LQ
Epson Printer
LaserWriter
Latest Sys. Files (No Finder)
Latest System Files
Local Network Startup
Network Printer Namer

Update Selection

Hard

Application Folder Selection

V1.2

installation process. However, screen prompts will guide you and will insure that you insert the correct disk in the drive. Installation is easy, but is more convenient if you have a second drive. Owners of two 3.5-inch drives should insert /SYSTEM.DISK in their primary drive and /SYSTEM.TOOLS in the second drive.

Updating GS/OS to 5.0.3

Owners of IIGS hard drive systems running under GS/OS 5.0 or later should follow these steps to update to 5.0.3:

1. Use any disk utility program except the Finder to determine if you have a file called "FINDER" in the System Folder on your drive. Rename that file "FINDER.2". (The procedures I describe will overwrite any file called "FINDER" in the System Folder.)
2. Boot your computer from the hard drive.
3. Insert the GS/OS 5.0.3 /SYSTEM.TOOLS disk in a drive and launch the Installer, which is in the root directory of the disk.
4. GS/OS will display the screen in *Figure 2*. Click on the "Disk" button if the Installer does not display the name of your boot volume over the right-hand window.
5. Scroll down the display box on the left side of the screen until you see "Latest Sys. Files (No Finder)" (see *Figure 3*) and click on it. (Apple did not update the Finder with version 5.0.3 and you should not replace your current version. This is particularly important if you use a program launcher such as EasyDrive, ProSel, or Wings. Clicking on "Latest System Files" will delete your launcher and replace it with the Finder.)
6. Click on the "Install" button to launch the installation procedure. Do not to click on the "Remove" button; that will remove all the system files from your hard disk without further warning!

You have now installed the system files and the A.C.E., AppleDisk 3.5, AppleDisk 5.25, and Direct Connect ImageWriter drivers and the standard CDevs on your hard disk. Next you must install the new SCSI driver and any other resources you need for your system. Proceed as follows:

1. Scroll to the SCSI Hard Drive choice on the left-hand window and click on that option to select that installer script. Then click on the "Install" button to install the SCSI driver.
2. Repeat step #1 for each of the following devices you have on your system: AppleShare, Apple-

Talk, AppleTalk ImageWriter/LQ, CD-ROM, Direct Connect ImageWriter LQ, LaserWriter, and UniDisk 3.5 (this is the white 3.5-inch drive used with Apple IIc computers).

If in doubt, install the driver. There is no harm in installing an unnecessary device driver. However, you will experience problems if you do not update a driver which should be updated.

3. Click on the "Quit" button to leave the Installer.
4. Reboot your system. You are now running under GS/OS 5.0.3.
5. If you changed the name of Finder in step #1 above, use your disk utility program to rename that file from "FINDER.2" to "FINDER".

Initial Installation on a Hard Drive

As you would expect, installing version 5.0.3 on a new drive or on a drive not already equipped with GS/OS 5.0 or later is a more complex process. It is complicated by the fact that the GS/OS /SYSTEM.DISK does not include the hard disk driver. Start by following these steps to install the SCSI driver on the /SYSTEM.DISK:

1. Make a copy of the GS/OS /SYSTEM.DISK.
2. Boot your computer with the copy of /SYSTEM.DISK. The computer will warn you that there is a SCSI device online. Ignore the warning and press the Return Key to continue the boot process.
3. Insert the GS/OS 5.0.3 /SYSTEM.TOOLS disk in a drive and launch the Installer, which is in the root directory of the disk. (If you have two 3.5-inch drives, insert /SYSTEM.TOOLS in the second drive. If you have a single drive, follow the prompts to swap disks.)
4. Scroll to "SCSI Hard Disk" in the left hand box and click.

A First Look at HyperCard GS

Apple IIGS developers have always concentrated on writing software that can run on minimally equipped systems. That expands the potential market for their products but constrains the development of powerful applications for the IIGS. HyperCard GS is about to change this situation. After using the beta version of this product, I believe that HyperCard GS may be the most exciting 16-bit program ever released for the IIGS.

HyperCard GS does everything you can do with the Macintosh version of HyperCard, and does it exactly the same way. (While authors will write books specific to HyperCard GS, there is no need to wait. The two products are so similar that you can use the standard HyperCard books to learn HyperCard GS. My favorite is Danny Goodman's *The Complete HyperCard 2.0 Handbook*, but there are many others books from which to choose.) In fact, my examination of a HyperCard GS stack with a block editor suggests that HyperCard and HyperCard GS stacks might be compatible with each other, although I could not test that compatibility in my preliminary examination of the product.

Just as Applesoft opened the Apple II to a large community of users, so will HyperTalk (the scripting language built into HyperCard GS) open the IIGS to an even larger community. HyperTalk is the most English-like of any programming language. There is no need to decipher GOSUB or GOTO structures; the program lets you link scripts to specific buttons on specific cards. If you want to modify the function of a button, its script is there for you to examine and change. The editing tools are built-in and are much more friendly than the rudimentary "editor" supplied with Applesoft. You can get excellent sound effects, create exciting graphic displays, and enhance the functionality of any open stack, all without understanding anything about the IIGS "Toolbox", object oriented programming, or program structure and flow.

There is a cost for all this power. The beta version of the HyperCard GS main system file which loads into memory exceeds 600K, and the total package of system files occupies almost two megabytes of space on a hard disk. A single powerful stack can exceed 500K, and you will no doubt acquire many stacks as you explore the new worlds which HyperCard GS opens. Thus, you cannot run HyperCard GS without expanded memory and a hard disk; you will want at least two megabytes of RAM and a 40-megabyte hard disk or larger. Although an accelerator is not essential, HyperCard GS is memory intensive and benefits greatly from IIGS accelerator products.

The IIGS is not the most powerful hardware platform available today; but neither is the Macintosh. When adequately enhanced, either machine is capable of running complex software which facilitates and enriches the life and work of its owner. HyperCard GS clearly demonstrates that Apple believes a significant number of IIGS owners have either upgraded their systems or will upgrade them to more powerful configurations to use this product. The rest is up to the IIGS community.

—*John Link*

5. Insert the /SYSTEM.DISK in a drive and click on the "Disk" button until /SYSTEM.DISK appears at the top of the right hand box.
6. Click on the "Install" button and respond to any prompts that appear on your screen.

Now that you installed the SCSI driver on your copy of GS/OS, you can install GS/OS on your hard drive. Proceed as follows:

1. Reboot your computer with the updated copy of /SYSTEM.DISK, and launch the Installer from /SYSTEM.TOOLS.
2. Click on the "Disk" button until the boot volume on your hard disk appears above the right-hand window.
3. Scroll to "Latest System Files" in the left-hand window and click once. (This option will install the system files and the Finder as your START file. That will take you to the Finder when you boot your computer. Follow the directions that came with your program launcher if you want to install EasyDrive, ProSel-16, Wings, or another launcher instead of the Finder.)
4. Click on the "Install" button to install the standard system files.
5. Scroll to "SCSI Hard Drive" in the left-hand window. Then click to highlight that choice and press the "Install" button to install that driver.
6. Scroll to "Advanced Disk Utility", click to highlight that choice, and press the "Install" button to install the utilities that let you format and partition your hard drive. *[Ed: Be careful if you ever use these utilities. They can delete all your data from the drive.]*
7. Repeat step #6 for each of the following devices on your system: Apple MIDI interface, AppleShare, AppleTalk, AppleTalk ImageWriter/LQ, Card 6850 MIDI, CD-ROM, Direct Connect ImageWriter LQ, Epson Printer, Laserwriter, Network Printer Namer (if you have an AppleTalk LaserWriter), UniDisk 3.5, and VideoMix.

   Install any device drivers you think you might need; you are better off having an unnecessary driver on your system than missing a driver you need.
8. Click on the "Quit" button to leave the Installer.
9. Reboot your system from the hard disk. You are now using GS/OS 5.0.3.

Installing on a Floppy Disk

Although GS/OS is at its best when run from a hard drive, many IIGS owners run GS/OS successfully from a 3.5-inch floppy drive. Any program that runs under GS/OS 5.0 or later should run under GS/OS 5.0.3, with one important exception: Do not try to install GS/OS on any copy-protected software.

Because each new release of GS/OS requires more disk space than the earlier versions, you may have problems finding enough room for it on older disks. In addition, early software may not be compatible with GS/OS 5.0.3. To confirm that a program runs under GS/OS 5.0.3, boot from the GS/OS 5.0.3 /SYSTEM.DISK into the Finder, and launch the older program. If it runs and prints without problems, it is probably compatible with GS/OS 5.0.3.

There are two ways to use GS/OS with floppy disk systems. One approach is to boot your computer with the GS/OS /SYSTEM.DISK and launch any compatible older program from the Finder. This is usually the easiest solution for owners of two 3.5-inch drives.

GS/OS with One 3.5-inch Drive

Owners of a single 3.5-inch drive who want to use GS/OS 5.0.3 should replace the operating system on each program disk with GS/OS. That will let you use your program disk to boot your computer.

First you must determine if there is enough room on the disk for the GS/OS 5.0.3. Boot your computer with the /SYSTEM.DISK and swap that disk with the /SYSTEM.TOOLS disk. Then launch the Installer and try to install "Latest Sys. Files (No Finder)". The Installer will try to install the system on your disk. If there is not enough room, an error message will tell you how much space you must create before you can install version 5.0.3.

I will describe how to install GS/OS 5.0.3 on AppleWorks GS 1.1 (which is compatible with GS/OS; older versions of AppleWorks GS are not

GS/OS compatible); that will serve as a paradigm for the process to follow to install GS/OS on other program disks. I will assume that you know how to use the Finder.

Follow these steps:

1. Boot from the GS/OS 5.0.3 /SYSTEM.DISK.
2. Insert a copy of /AWGS.SYSTEM and delete /AWGS.SYSTEM/SYSTEM/START by dragging it to the trash. This makes room for the new files you will install.
3. Insert the /SYSTEM.TOOLS disk and launch the Installer.
4. If necessary, reinsert /AWGS.SYSTEM into a drive and click on the "Disk" button until its name appears over the right-hand window on the screen.
5. Select "Latest Sys. Files (No Finder)" from the choices in the left-hand window on the screen and click on the "Install" button. That will install the new system files but will not add the Finder.
6. Click on the "Quit" button to leave the Installer.
7. Drag the following unnecessary files to the trash:

```
/AWGS.SYSTEM/
        ..BASIC.SYSTEM
        ..BASIC.LAUNCHER
/AWGS.SYSTEM/FONTS
        ..COURIER.10
        ..HELVETICA.10
        ..TIMES.10
```

8. Decide whether you want to use the new version 5.0.3 ImageWriter driver or the one furnished by Claris. Then drag *either* of the following files to the trash:

```
/AWGS.SYSTEM/DRIVERS
        ..IMAGEWRITER.CL  (Claris special Image-
                           Writer driver)
        ..IMAGEWRITER     (New Apple ImageWriter
                           driver)
```

9. Now you must delete an additional 41 blocks from the disk to make room for the Finder. You can choose from any of the following, as long as the block count totals 41:

```
/AWGS.SYSTEM/SYSTEM/CDEVS        (Deleting these
        ..ALPHABET       (10)     disables access
        ..DIRECTCONNECT  (11)     via the graphics
        ..GENERAL        ( 8)     Control Panel.
        ..KEYBOARD       (10)     You may still
        ..MODEM          (12)     access all these
        ..MONITOR        (11)     functions from
        ..MOUSE          ( 8)     the text Control
        ..PRINTER        (12)     Panel.)
        ..RAM            (16)
        ..SLOTS          (13)
        ..SOUND          (10)
        ..TIME           (17)
/AWGS.SYSTEM/SYSTEM/SYSTEM.SETUP
        ..TS2            (82)      (Disables boot-
                                    ing on ROM 1
                                    machines)
        ..TS3            (30)      (Disables boot-
                                    ing on ROM 3
                                    machines)
```

10. Drag START from /SYSTEM.DISK/SYSTEM to /AWGS.SYSTEM/SYSTEM. This re-establishes the Finder as the program launcher. AppleWorks GS 1.1 is now ready to boot into GS/OS 5.0.3.

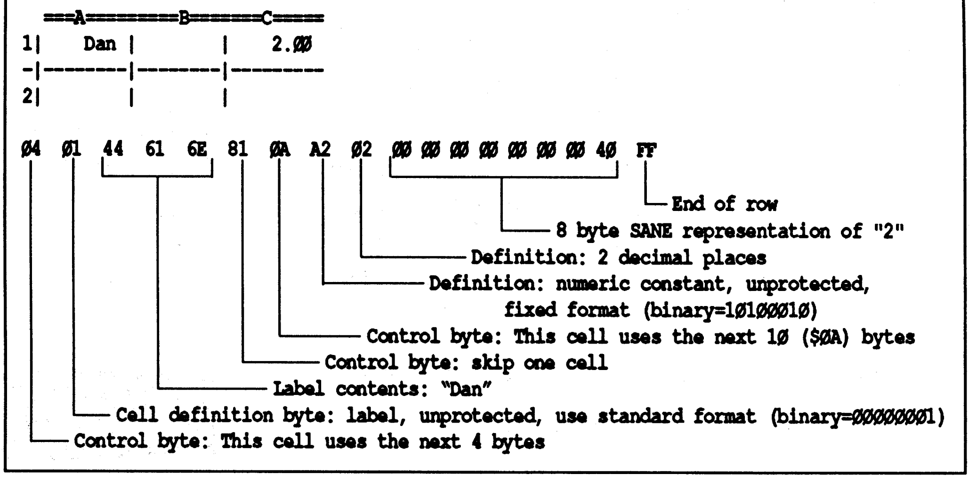
Spreadsheet Data Structure — Part 1

by Dan Verkade

The AppleWorks spreadsheet uses the most complex data structure of all three AppleWorks modules. This is the first of two articles that describe the basics of that structure. The author assumes that you read the articles entitled "Limitations of the AppleWorks Spreadsheet" and "Memory Management" in the October 1987 and August 1990 issues of the ***AppleWorks Forum****.*

Of all the AppleWorks modules, the spreadsheet module does the most magic with your data. AppleWorks' spreadsheet makes it easy to manipulate rows and columns of data, and control the format and contents of individual cells. The program automatically changes all affected cells when you make a change to any other cell.

Figure 1: Sample Spreadsheet Record



```
 ===A========B=======C=====
1|    Dan |        |   2.ØØ
-|--------|--------|--------
2|        |        |

Ø4  Ø1  44  61  6E  81  ØA  A2  Ø2  ØØ ØØ ØØ ØØ ØØ ØØ ØØ 4Ø  FF
```

End of row
8 byte SANE representation of "2"
Definition: 2 decimal places
Definition: numeric constant, unprotected,
fixed format (binary=1Ø1ØØØ1Ø)
Control byte: This cell uses the next 1Ø ($ØA) bytes
Control byte: skip one cell
Label contents: "Dan"
Cell definition byte: label, unprotected, use standard format (binary=ØØØØØØØ1)
Control byte: This cell uses the next 4 bytes

A spreadsheet program requires many lines of complex computer code and a concise, well-defined data structure. The efficient data structure allows more room for data, requires fewer instructions to manipulate the data, and speeds up the program's operation.

This is the first of two articles in which I will describe the data structures used by AppleWorks' spreadsheet module. This month I will describe how the AppleWorks spreadsheet module stores labels, values, and formulas and how the program manages cell references. Next month I will describe how AppleWorks arranges its data so you can insert rows and columns, and how it handles recalculations.

The Basics

AppleWorks stores each row of spreadsheet data as a separate record on the desktop. The program then divides each row record into separate cells. For example, a spreadsheet record (row) might look like the example in *Figure 1.*

The data for each cell begins with a control byte which indicates one of three conditions:

1. A value between 1-127 ($Ø1-$7F) indicates the number of bytes used for that cell.
2. A value between 129-254 ($81-$FE) indicates that there are empty cells to skip. Subtract 128 ($8Ø) from this number to determine the number of empty cells.
3. A value of 255 ($FF) indicates that this is the last cell in the row.

Note the implications of the first byte of the data structure:

1. A cell cannot hold more than 127 characters of data.
2. There can be no more than 127 columns in any spreadsheet row.

Following the control byte is the "cell definition byte" that tells AppleWorks about the contents of the cell. The first three bits of this byte indicate

A Quick Lesson in Binary Mathematics

Here are some rules to help you understand the operation of binary mathematics:

1. A single byte can contain any value in the range of Ø - 255.
2. Each byte consists of eight bits.
3. Each bit acts like a switch with two possible states; "on" (a value of "1") or "off" (a "Ø").
4. Each switch is numbered consecutively in order of significance from left to right, starting with seven (left-most) and ending with zero (right-most).
5. If the switch is "off", that bit takes the value of Ø.
6. If the switch is "on", that bit takes on the value of 2 raised to the power of the number of the switch. For example, when switch three is on, the value of that bit will be 2 raised to the 3rd power, or 8.
7. The value of the byte is the sum of the values of the bits. For example, if switches 6, 4, and 2 are on, those bits have the values of 64, 16, and 4, respectively. The other switches are off, and have the value of Ø. The value of the byte is the sum of 64, 16, and 4, or 84.
8. No two combinations of bits generate the same number. Once you know the value of a byte, you can determine the state of each switch in that byte. For example, consider the example of the switch settings in *Figure A*.
9. When you set a switch (bit) on, you add its value (2 raised to the power of its number) to the byte value. When you clear a switch (turn it off), you subtract its value from the byte. For example, consider the switch settings in *Figure A*. If you turn on switch 1, you add 2 to the byte value, and make it 199. Turn that switch off again and the byte value is 197.

If this set of switches represents a cell definition byte, the cell contains a formula without protection that should be displayed as a percent. Thus, every cell with a definition byte value of 197 contains formulas without protection that should be displayed as a percentage.

—Dan Verkade

Figure A: Cell Definition Byte Switch Settings

switch number:	7		6		5		4		3		2		1		0		
decimal value:	128		64		32		16		8		4		2		1		
On:	X		X								X				X		
Off:					X		X		X				X				
	128	+	64	+	0	+	0	+	0	+	4	+	0	+	1	=	197

whether the cell contains a label, a formula, or a number as follows:

Bit: 7 6 5	Cell Contents:
0 0 0	Label
0 0 1	Repeated label
1 0 1	Numeric constant
1 1 0	Formula
1 1 1	Blank with a value of Ø

The next two bits specify the type of protection on the cell.

Bit: 4 3	Protection
0 0	No protection, anything may be typed into the cell
0 1	No numeric constants or formulas may be typed into the cell
1 0	No labels may be typed into this cell
1 1	Nothing may be typed into this cell

Remember that AppleWorks respects these protection bits only when global protection is active.

The last three bits specify the format of the cell. The format depends on whether the cell contains text or a value. AppleWorks ignores these bits if the cell contains a repeated label.

If the cell contains a numeric entry:

Bit: 2 1 0	Format:
0 0 1	Use spreadsheet standard
0 1 0	Fixed
0 1 1	Dollars
1 0 0	Commas
1 0 1	Percent
1 1 0	Appropriate

If the cell contains text:

Bit: 2 1 0	Format:
0 0 1	Use spreadsheet standard
0 1 0	Left justify
0 1 1	Right justify
1 0 0	Center

Figure 2: Tokens for Functions and Operators

$CØ	@DEG	$D6	@PMT	$EC	<> (not equal to)
$C1	@RAD	$D7	@TERM	$ED	>= (greater than or equal to)
$C2	@PI	$D8	@RATE	$EE	<= (less than or equal to)
$C3	@TRUE	$D9	@ROUND	$EF	= (equals)
$C4	@FALSE	$DA	@OR	$FØ	> (greater than)
$C5	@NOT	$DB	@AND	$F1	< (less than)
$C6	@ISBLANK	$DC	@SUM	$F2	, (comma)
$C7	@ISNA	$DD	@AVG	$F3	^ (exponentiation sign)
$C8	@ISERROR	$DE	@CHOOSE	$F4	) (right parenthesis)
$C9	@EXP	$DF	@COUNT	$F5	- (minus)
$CA	@LN	$EØ	@ERROR	$F6	+ (plus)
$CB	@LOG	$E1	@IRR	$F7	/ (divide)
$CC	@COS	$E2	@IF	$F8	* (multiply)
$CD	@SIN	$E3	@INT	$F9	((left parenthesis)
$CE	@TAN	$E4	@LOOKUP	$FA	- (unary minus eg., -A3)
$CF	@ACOS	$E5	@MAX	$FB	+ (unary plus eg., +A3)
$DØ	@ASIN	$E6	@MIN	$FC	... (dots)
$D1	@ATAN2	$E7	@NA	$FD	Next 8 bytes are SANE numeric
$D2	@ATAN	$E8	@NPV	$FE	Next 3 bytes are column, row reference
$D3	@MOD	$E9	@SQRT	$FF	Next n bytes are a string preceded by length
$D4	@FV	$EA	@ABS		
$D5	@PV	$EB	Not Used		

The cell definition byte is followed by another definition byte whose meaning depends on the type of data in the cell. If the cell is a numeric constant, the first bit is "off" (i.e., equal to zero) and AppleWorks ignores the next four bits. If the cell is a label, this byte contains the first character of text, and the length of the text is indicated by the control byte. If the cell is a repeated label, this byte contains the repeated character.

The first bit is always on if the cell contains a formula. The next four bits indicate the following:

```
Bit: 6  Last evaluation of this formula was @NA.
     5  Last evaluation of this formula was @ERROR.
     4  Cell needs to be recalculated. (AW 2.1 and
           later)
     3  The formula returned a string value, not a
           numeric value. (AW 3.0 only.)
```

The last three bits tell AppleWorks how many decimal places are available for those formats that require it. Note that three bits allow from zero to seven decimal places because there are seven possible combinations of on's and off's when you have three binary "switches".

The bytes after the definition bytes contain data as follows:

Cell Content	Stored Data
Repeated label	No additional data.
Label	The text appears here using as many bytes as necessary to complete the string.
Numeric constant	Contains 8 bytes. AppleWorks uses the Standard Apple Numeric Environment (SANE) to store all spreadsheet numeric values.

Implications

The decision to use SANE for spreadsheet calculations has two ramifications. First, any numeric value, no matter how large or small, requires 11 bytes if standing alone in a cell or 9 bytes if in a formula. Second, although AppleWorks displays numeric values with up to seven decimal places, the program uses the full precision available in all calculations. Thus, the displayed and underlying numbers in calculations often do not match. You can change the precision of the displayed numbers by modifying the cell format. You can also use the @ROUND function to decrease the precision of the underlying calculated number so it matches the displayed value.

AppleWorks calculates and stores all numbers in 8-byte SANE format but displays all numbers in decimal format. The conversion between formats takes time. That is why the spreadsheet scrolls more slowly through a page of numbers than through a page of text.

Formulas

Several different arrangements can occur if the cell contains a formula. First, the bytes following the second definition byte describe the results of the last evaluation of this formula. If the formula evaluates as a number, the next eight bytes contain the number in SANE format. If it evaluates as a string, the next byte is the length of the string and the following bytes contain the data. This data is only as accurate as the last recalculation.

Next comes the formula itself. AppleWorks stores the formula in "token format", not as it appears on the screen. A token is a one-character replacement for a function or operator inside the formula. For example, the token for @CHOOSE is $DE. Thus, you will find $DE in the memory block that stores a formula containing an @CHOOSE function. Tokens also tell AppleWorks to expect a number, string, or cell reference within a formula.

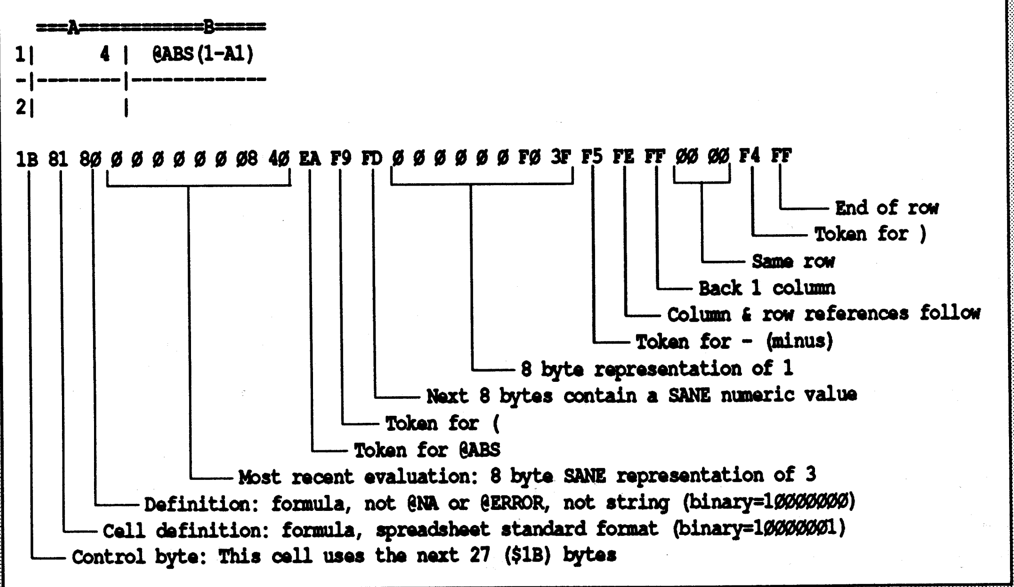
AppleWorks uses tokens to save memory and to increase the speed of calculations. Tokens eliminate the need for the computer to translate each function and operation from English into "computerese" every time it recalculates the spreadsheet. Instead, AppleWorks translates each operation into tokens when you first enter the formula.

Figure 2 contains the tokens for all the functions and operators.

Figures 3A and *3B* depict sample data from cells containing formulas that yield numeric and non-numeric results.

Figure 3: Cell Forumulas

Figure 3A: Cell Formula that Yields Numeric Results

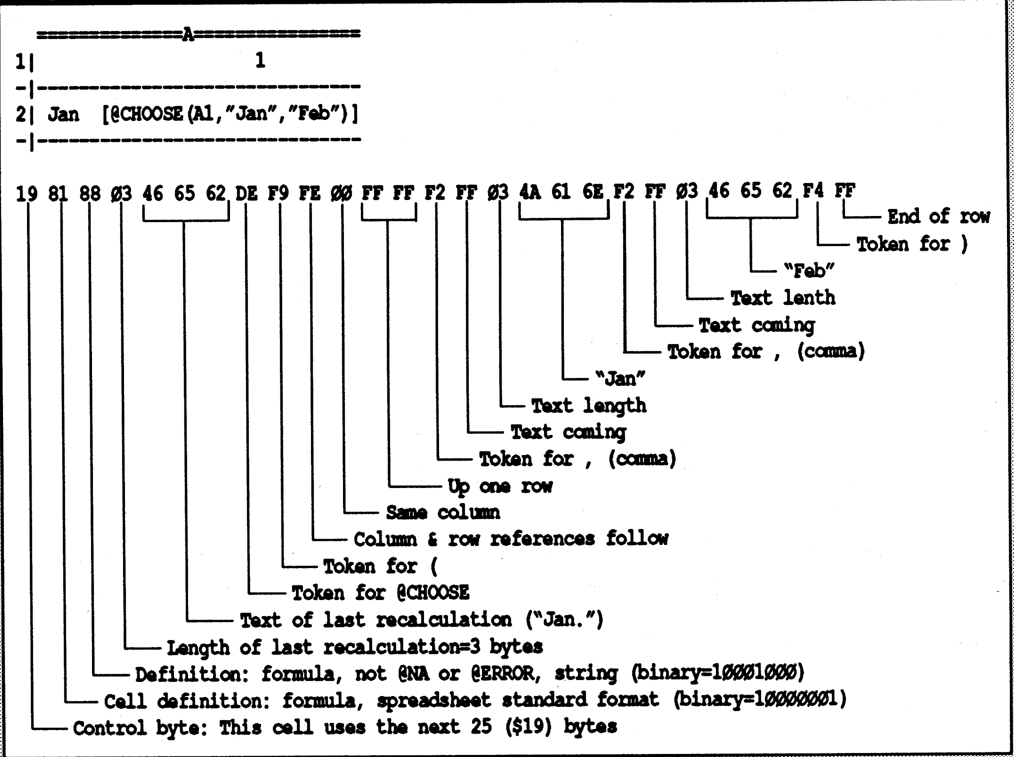


```
 ===A==========B=====
1|      4 |  @ABS(1-A1)
-|--------|------------
2|        |

1B 81 8Ø Ø Ø Ø Ø Ø Ø Ø8 4Ø EA F9 FD Ø Ø Ø Ø Ø Ø FØ 3F F5 FE FF ØØ ØØ F4 FF
```

End of row
Token for)
Same row
Back 1 column
Column & row references follow
Token for - (minus)
8 byte representation of 1
Next 8 bytes contain a SANE numeric value
Token for (
Token for @ABS
Most recent evaluation: 8 byte SANE representation of 3
Definition: formula, not @NA or @ERROR, not string (binary=1ØØØØØØØ)
Cell definition: formula, spreadsheet standard format (binary=1ØØØØØØ1)
Control byte: This cell uses the next 27 ($1B) bytes

Figure 3B: Cell Formula that Yields Text Results



```
 ============A===============
1|                    1
-|---------------------------
2| Jan  [@CHOOSE(A1,"Jan","Feb")]
-|---------------------------

19 81 88 Ø3 46 65 62 DE F9 FE ØØ FF FF F2 FF Ø3 4A 61 6E F2 FF Ø3 46 65 62 F4 FF
```

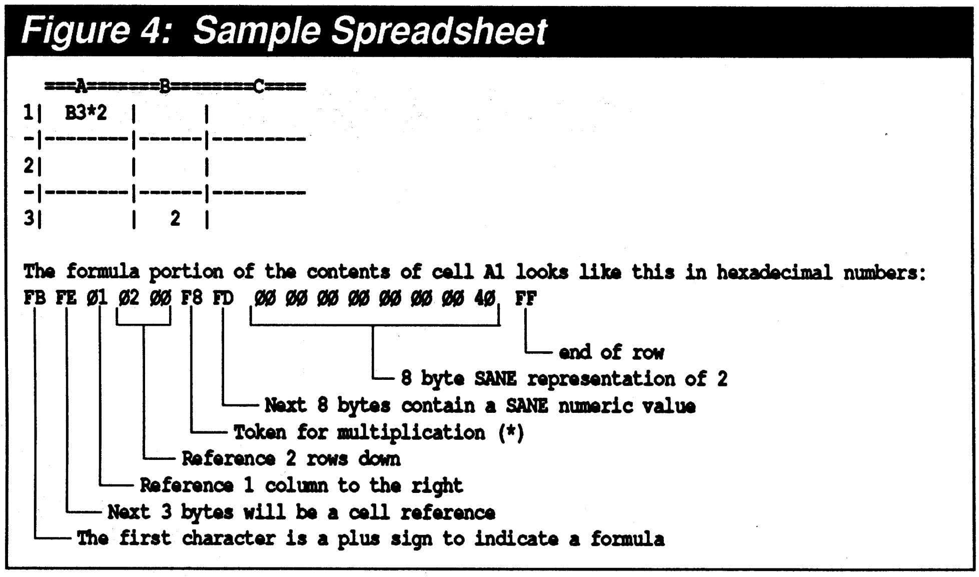
End of row
Token for)
"Feb"
Text lenth
Text coming
Token for , (comma)
"Jan"
Text length
Text coming
Token for , (comma)
Up one row
Same column
Column & row references follow
Token for (
Token for @CHOOSE
Text of last recalculation ("Jan.")
Length of last recalculation=3 bytes
Definition: formula, not @NA or @ERROR, string (binary=1ØØØ1ØØØ)
Cell definition: formula, spreadsheet standard format (binary=1ØØØØØØ1)
Control byte: This cell uses the next 25 ($19) bytes

Relative Cell References

Most users appreciate the spreadsheet's ability to maintain relative references while copying formulas. Upon studying the data structure, you will find that maintaining relative cell references is quite simple.

AppleWorks stores the relationship between every referenced cell and the cell that contains a formula. For example, if a cell contains the formula +B3*2, AppleWorks does not reference cell B3 directly. Instead, it stores the number of rows and columns between the cell that contains the formula and the referenced cell, cell B3. When you display the formula, AppleWorks determines which cell appears in the formula by calculating the cell that is a specified distance from the current cell. It happens so quickly that it is easy to think that AppleWorks keeps track of every cell by an absolute reference.

Figure 4: Sample Spreadsheet



```
  ===A=======B=======C====
1|  B3*2  |      |
-|--------|------|---------
2|        |      |
-|--------|------|---------
3|        |  2   |

The formula portion of the contents of cell A1 looks like this in hexadecimal numbers:
FB FE Ø1 Ø2 ØØ F8 FD  ØØ ØØ ØØ ØØ ØØ ØØ ØØ 4Ø  FF
|  |  |  |__|  |  |   |_______________________|  |
|  |  |    |   |  |               |             |__ end of row
|  |  |    |   |  |               |__ 8 byte SANE representation of 2
|  |  |    |   |  |__ Next 8 bytes contain a SANE numeric value
|  |  |    |   |__ Token for multiplication (*)
|  |  |    |__ Reference 2 rows down
|  |  |__ Reference 1 column to the right
|  |__ Next 3 bytes will be a cell reference
|__ The first character is a plus sign to indicate a formula
```

For example, consider the spreadsheet in *Figure 4*. Imagine that you want to copy cell A1 into cells A2 through A6. The cell reference never changes; it always specifies the cell one column to the right and two rows down from the current cell. Thus, AppleWorks simply copies the relationship for each of the new cells that contain the copied formula.

It is interesting to note that AppleWorks does more work when you select "No change" than when you select "Relative". When you select "No change" AppleWorks must change each row or column reference depending on the direction of the copy to keep the relative reference correct.

Making Best Use of the 127 Bytes

I mentioned earlier that a cell can contain no more than 127 bytes. This includes the control byte, the definition bytes, the most recent evaluation, and the formula itself. As a result, it is possible to build text formulas that are so large that AppleWorks cannot display the text in its entirety.

For example, create a spreadsheet with column A 70 characters wide. Then fill row 1 of column A with 70 "X"s. Enter the formula:
`@CHOOSE(1,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12))`
in row 18. Notice that AppleWorks displays fewer than 70 "X"s for this formula. Remove the cell references from the formula and you will gain five "X"s for every deleted cell reference. You gain one byte for the comma, one byte for the reference token, one byte for the column reference and two bytes for the row reference.

Conclusion

This month, I described the basic elements of AppleWorks' spreadsheet data structure. Next month I will describe how AppleWorks arranges its data so you can insert rows and columns, and how it handles recalculations. I will also offer suggestions to help you use this information to improve your spreadsheet templates.

[Dan Verkade is the developer of TimeOut ReportWriter, DoubleData, and other popular AppleWorks enhancements.]

Late News and Special Offers

Apple Computer

Apple Computer recently announced two new technical publications; the Education Technical Bulletin (which replaces the company's Apple II Technical Notes and includes technical information of interest to educators about Apple II and Macintosh products) and the Macintosh Technical Bulletin. Each 40-page bulletin is published six times a year and costs $129 per year. Order these publications from your local Apple dealer.

Applied Engineering

Applied Engineering recently introduced the GS-RAM II, a low-cost Apple IIGS memory expansion card that increases system memory by 1-4 megabytes. The board is DMA compatible and comes with Applied's AW3 Expander and diagnostic utility programs. A 1-megabyte GS-RAM II lists for $179; a 2-megabyte card lists for $259; a 4-megabyte board lists for $419.

Applied also announced the availability of an Apple IIGS 3.5-inch floppy disk drive that stores up to 1.6 megabytes of data on a high density disk. The drive, which can read and write on both high density and standard disks, requires an Apple IIGS computer running GS/OS. Suggested retail price is $339.

Owners of Applied 3.5-inch drives can upgrade to the new high density format for $79. Upgrades are factory installed; contact Applied to arrange for the upgrade. *[Applied Engineering, Box 5100, Carrollton, Texas 75011; (214) 241-6060.]*

Beagle Bros

Beagle Bros recently released Outliner 1.2, an update to the company's outlining program that runs within AppleWorks. Outliner 1.2 offers a "Print to the clipboard" option that makes it easy to move outlines into the AppleWorks word processor. These outlines carry tab and margin settings that let you print attractively formatted documents. Version 1.2 also offers improved formatting of multi-line topic headings. Owners of earlier versions of Outliner (including owners of JEM's Outline 3.0) can get updates to version 1.2 from any of NAUG's Beagle Buddies (see page 22 of last month's issue of the ***AppleWorks Forum***).

Beagle also announced the release of TextTools 1.31, a maintenance release of the company's AppleWorks word processor enhancements. TextTools 1.31 fixes problems discovered in MarkMerge and SuperFind shortly after the release of versions 1.2 and 1.3. (TextTools 1.31 includes MarkMerge 1.3 and SuperFind 1.2.)

NAUG members who bought TextTools from any source can update to version 1.31 directly from NAUG. Send NAUG your original disk and a return mailer with postage. We would appreciate, but do not require, a $1 donation to help reimburse NAUG for the expenses associated with this disk exchange program. Our thanks to Beagle for supplying the version 1.31 disks for this exchange.

FrankSoft Publishing

FrankSoft Publishing Company recently released three sets of AppleWorks spreadsheet templates that help users manage their personal finances.

Asset Analysis is an investment and asset analysis template that tracks stocks, mutual funds, bonds, IRAs, pensions, certificates of deposits and other assets and liabilities. The template prints summaries for each class of asset and generates separate reports for your IRAs, pension accounts, life insurance, and other assets. Documentation, including suggestions, shortcuts, and examples, appears in a 20-page word processor file on the disk.

Asset Analysis 1.1.1 works with AppleWorks 1.3 and later and requires a 53K desktop. Version 2.1.1, which integrates the different portfolios and reports, requires AppleWorks 3.0 and a 133K desktop.

Your Net Worth produces a net worth statement that is accepted by most financial institutions. The template produces an asset analysis, a current asset performance statement that reports the percent gain or loss for each investment, and a past asset performance statement. Your Net Worth requires Apple-

Works 2.1 or later and a 40K desktop.

Retirement Planner helps plan and track your retirement investments. The "Accumulation" segment of the template tracks the current and future value of retirement savings and investments. The "Withdrawal" segment computes the impact of different withdrawals and earnings during the retirement years. Retirement Planner requires AppleWorks 2.x or later and runs on a 40K desktop.

Until March 31, 1991, **NAUG** members can get substantial discounts on FrankSoft products. The discount prices are as follows:

	List Price	NAUG Price
Asset Management 1.1.1	$39.95	$19.95
Asset Management 2.1.1	49.95	24.95
Your Net Worth	34.95	17.50
Retirement Planner	34.95	17.50

Include $3 s/h per order. Illinois residents should add 6.25% sales tax. The company accepts MasterCard and Visa and offers **NAUG** members "satisfaction guaranteed or your money back". *[FrankSoft Publishing, 3300 33rd Avenue, Rock Island, Illinois 61201; (309) 788-7663; Fax: (309) 788-7664.]*

JEM Software

JEM Software offers **NAUG** members a special discount to introduce the company's new credit card telephone order line. While supplies last, **NAUG** members can get Platinum Paint, Beagle Bros' new graphics program for the Apple IIGS, for $50 plus $2 s/h (Platinum Paint retails for $99.95). This is a limited edition of Platinum Paint signed by the author Matt Reimer. JEM will only accept Visa and MasterCard telephone orders at this special price. Identify yourself as a **NAUG** member when you leave a message on the company's telephone order line. *[JEM Software, 7578 Lamar Court, Arvada, Colorado 80003; (303) 422-4856.]*

New Concepts

NAUG members who have difficulty reading the Apple II screen will appreciate the Compu-Lenz, a 9" x 15" magnifying lens that mounts in front of the Apple display and significantly enlarges the image on the screen. The Compu-Lenz has a suggested list price of $204.95; an optional swivel mount retails for $20.95. Until April 1, 1991 **NAUG** members can buy the Compu-Lenz directly from New Concepts for $163.96 and a swivel mount for $16.76. Add $4.95 s/h for the Compu-Lenz and an additional $4.95 for the swivel mount.

Our thanks to Alan Kahn for bringing the Compu-Lenz to our attention. Mr. Kahn recommends the Compu-Lenz to his fellow sight-impaired **NAUG** members. *[New Concepts, 6710 Embassy Boulevard, #204, Port Richey, Florida 34668; (813) 845-7544.]*

Quality Computers

Quality Computers recently announced the release of "Ruth Witkin's Best New AppleWorks Templates". Templates include an investment planner, a banking recordkeeper, a combined balance sheet and income statement, a sales analyzer, an income tax and FICA tax calculator, a complete employee information file, and a rotary index card printer. Ruth Witkin's Best New AppleWorks Templates has a suggested list price of $39.95.

Until February 15, **NAUG** members can buy the Ruth Witkin templates, RepairWorks, or SuperPatch 7.0 directly from Quality for $29.95. Any two of these products cost $50. Quality now offers **NAUG** members a 30-day money back guarantee on their software products. *[Quality Computers, Box 665, St. Clair Shores, Michigan 48080; (800) 443-6697.]*

Zip Technology

Until March 1 **NAUG** members can trade in any Zip, Applied Engineering, or RocketChip accelerator and take an additional discount from Zip's special **NAUG** member prices. Prices, including trade-ins, are as follows:

Product	List Price	NAUG Price	Trade in	Final NAUG Price
4-meg. Zip Chip*	$125	$94	-$15	$79
8-meg. Zip Chip*	199	139	- 40	99
Zip GSX†	350	239	- 40	199

*For Apple IIe, IIc, IIc Plus, and Laser 128
†For Apple IIGS

To get these special prices you must order directly from Zip Technology and identify yourself as a **NAUG** member. *[Zip Technology, 5601 West Slauson Avenue, Suite 190, Culver City, California 90230; (213) 337-1313.]*

We don't recommend plucking your Vulcan™ off a table and dragging it along the ground.

But it's nice to know you could.

October 30, 1990

Gentlemen:

I use my Apple IIGS with a Vulcan and an AE A/D converter at professional waterski tournaments to measure jump distances. I wish to commend you on the durability of your Vulcan Hard Drive.

I was using the computer at the U.S. National Waterski Championships in August when a gust of wind picked up the canopy under which we were operating. Wires running to my system were attached to the canopy and when the canopy blew away, it pulled the computer with the Vulcan off the table and onto the ground, dragging it along the ground some twenty feet. I was actually in the process of writing data to the hard disk at the time.

Even though the incident pulled many of the attaching wires out of the computer, no damage occurred to the computer or the hard drive. I subsequently verified all of the data on the hard drive and found no errors and no bad or damaged blocks in either ProDOS or MS-DOS (I use half my storage for my PC Transporter's MS-DOS files and half for ProDOS).

Needless to say, I am very grateful for a soundly built and well-engineered product.

Sincerely yours,
Roger Dilling
Milledgeville, Georgia

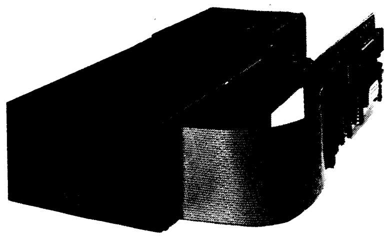
We hope you never drag your hard disk through the dirt, but can appreciate the engineering required to make the above letter possible. Until recently, if you wanted a hard disk for your Apple, you had to add an outdated, boxy external to your desktop clutter. Now, with Vulcan™ on the scene, you have an internal to consider. One that's lightning fast, clean, powerful and affordable.

A glance at the other computer manufacturers; IBM, Compaq, Dell, Mac, tells you something. They're all very different systems, but all come with internal hard disks (it's hard to even find a Mac these days without an internal hard disk). The reason? Internals are the latest advance. The modern storage solution. They become a transparent part of your system, and in the case of Vulcan, actually enhance the rest of your system.

Enhancing the rest of your system. Many feel Apple's standard power supply is insufficient. Add a Vulcan and you make a significant improvement to the rest of your system. The high efficiency power supply in Vulcan is rated in excess of 70 watts, nearly double the capacity of Apple's standard power supply and that of other drives. Vulcan power supply components are heatsinked to the aluminum case for cool operation and long life. And we added an ultra-quiet, flush-mounted cooling fan to keep things cool inside. Vulcan actually beefs up your power supply. External hard disks drain it.

Ease of use. Most hard disks are pretty intimidating. It's frustrating to bring home a new hard disk, only to discover you've got to spend a lot of time setting up for your particular operating system, partitions and formatting. Vulcan comes pre-formatted, pre-partitioned and includes the latest Apple Operating System. You'll be using your Vulcan within minutes. Just pull out your old power supply, plug the Vulcan in it's place, insert the card and turn on the computer. Vulcan will boot to your familiar Apple Finder in a few seconds. Now *that's* ease of use.

Incider/A+ Magazine put it simply in their "Best of the Best" Holiday Shopping Guide: *"The best internal hard disk is the Vulcan from Applied Engineering - you can use it with DOS 3.3, ProDOS or GS/OS, and it comes with its own fan and power supply".* Vulcan incorporates the most popular standard protocols for a hard disk and includes an ultra-fast 16-bit data bus controller, not the less expensive 8-bit others use. And since Vulcan is fully compatible with our PC Transporter, you can create ProDOS or MS-DOS storage space, in addition to GS/OS, DOS 3.3, CP/AM or Pascal 1.3. No other hard disk offers you greater flexibility.



Vulcan gives your Apple IIGS, IIe, II+ or II upgradeability (20-100 MB), useable speed, safe power (its FCC certified and works on 110/220 VAC, even European 115/230 VAC at 50-60 Hz), practically unlimited size and AE's famous tech support and one-year warranty.

Order today! To order or for more information, see your dealer or call (214) 241-6060 today, 9 am to 11 pm, 7 days. Or send check or money order to Applied Engineering. MasterCard, VISA and C.O.D. welcome. Texas residents add 8 1/4% sales tax. Add $10 outside U.S.A.

Vulcan 20 MB	**$699**
Vulcan 40 MB	**$899**
Vulcan 100 MB	**$1795**

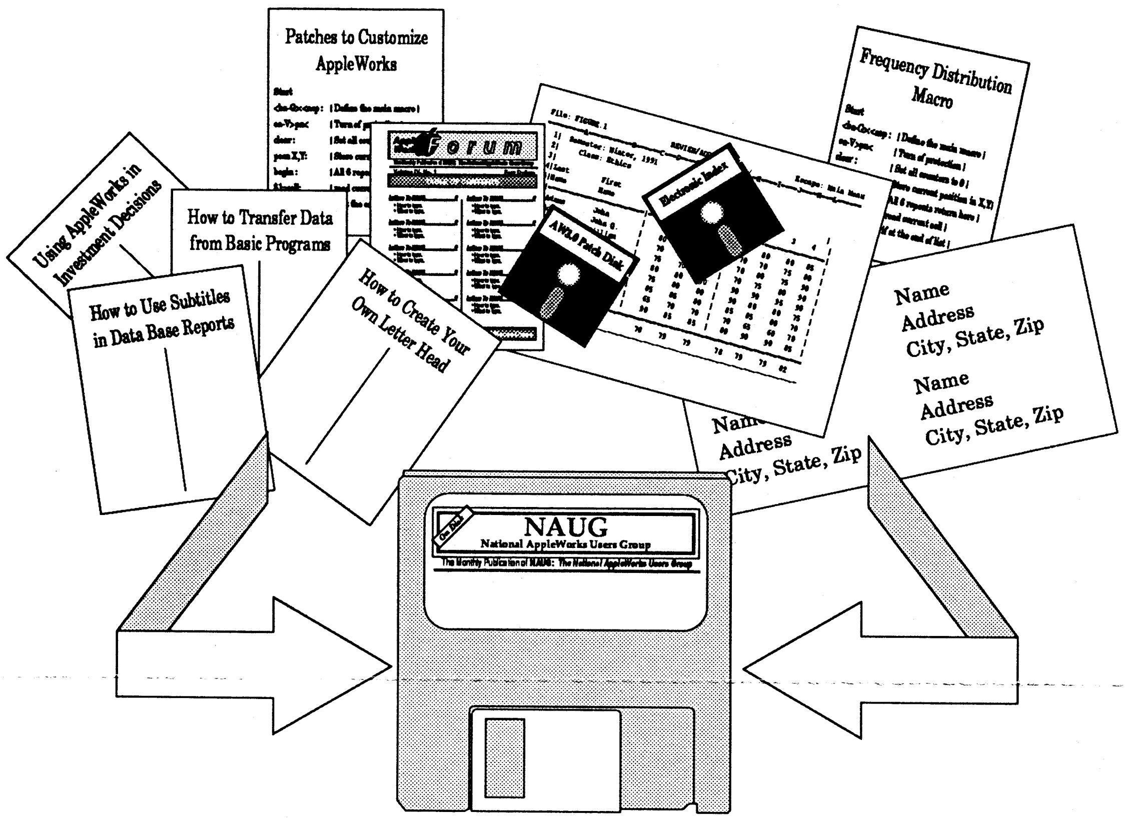
ÆE

Applied Engineering®
The enhancement experts.
A Division of AE Research Corporation.

(214) 241-6060

P.O. Box 5100
Carrollton, TX 75011

Made IN THE USA


Patches to Customize AppleWorks
Frequency Distribution Macro
Using AppleWorks in Investment Decisions
How to Transfer Data from Basic Programs
How to Use Subtitles in Data Base Reports
How to Create Your Own Letter Head
Forum
Electronic Index
AW2.0 Patch Disk
Name
Address
City, State, Zip
Name
Address
City, State, Zip
Name
Address
City, State, Zip
NAUG
National AppleWorks Users Group
The Monthly Publication of NAUG: The National AppleWorks Users Group

NAUG on Disk.

Each monthly issue of NAUG on Disk includes:

- The *AppleWorks Forum*

  A complete electronic version of the ***AppleWorks Forum*** that you can read with AppleWorks or AppleWorks GS, and search with FastFind or TimeOut DirecTree. Also compatible with software for the visually impaired. Includes a printed copy of all figures from the newsletter. Easy to store and use.

- Electronic copies of all macros and patches

  All the macros and patches described in that month's issue of the ***AppleWorks Forum***. Eliminates the typing necessary to enter these files.

- Working templates

  Working versions of all the templates in that month's issue of the ***AppleWorks Forum***. You can use these templates directly from the disk or can adapt the files for your own applications or for use with students.

- Updates to the Electronic Index

  Updates to NAUG's Electronic Index, an AppleWorks data base file that makes it easy to find any article in the back issues of the ***AppleWorks Forum***.

- Public domain templates, utilities, and programs

  The new public domain templates, utilities, and programs added to the NAUG library that month. A convenient and inexpensive way to update your collection of AppleWorks templates, programs, and utilities.

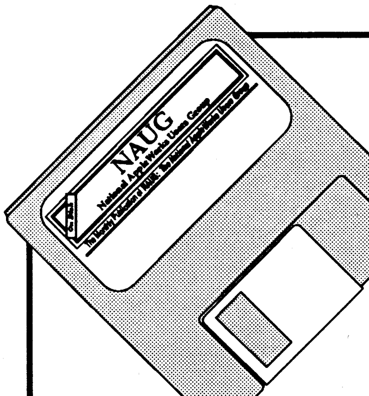
- Unpublished articles

  Articles not yet printed in the ***AppleWorks Forum***. These articles suggest techniques and ideas to help you enhance your AppleWorks productivity. Articles include a balance of items of interest to novice and advanced AppleWorks users.

- Members Helping Members data base

  An electronic version of NAUG's valuable Members Helping Members data base. You can use AppleWorks to search this file for member-volunteers who provide free telephone consulting to help you get answers to your AppleWorks questions.

— Satisfaction guaranteed or your money back. —


NAUG
National AppleWorks Users Group
The Monthly Publication of NAUG: The National AppleWorks Users Group
On Disk

NAUG on Disk*

How to get NAUG on Disk:

Single Disks: Single back issues are available at $8.00 per month. (Price includes shipping to the U.S., Canada, or Mexico.)

Subscription Information: You can upgrade your subscription to add NAUG on Disk to your monthly membership. (Multiply the number of months remaining in your subscription by $7.00.)

The Complete 1990 Collection: The entire 1990 collection costs $90.00.

Disk Contents:

January 1990

January AppleWorks Forum
6 Unpublished Articles
MHM Data Base
Data Base Address File Template
Tables Format Report Template

February 1990

February AppleWorks Forum
AppleWorks 3.0 Patch Disk v. 1.5
FormsWorks 01 Templates
FormsWorks 02 Templates
13 Patches to AppleWorks 3.0
Data Base Address File Template
Labels Format Report Template
Program to Produce Letterhead
Program to Produce Address Labels

March 1990

March AppleWorks Forum
Bank Sizer Program
MicroBio/MicroChem Dictionary
8 Patches to AppleWorks 3.0
Spreadsheet Chart Template
Spreadsheet Table Template
Spreadsheet Budget Template
Spreadsheet Gradebook Template
Spreadsheet What If? Template
"Twister" Program

April 1990

April AppleWorks Forum
AppleWorks Footnote System Macros
NAUG Home 01 Templates
Payroll Calculator 1990 Template
10 Patches to AppleWorks 3.0
Spreadsheet Column Expander Macro
Spreadsheet Column Shrinker Macro

May 1990

May AppleWorks Forum
Ram Disk Tutor Files
Macro Library Integrator Files
Spreadsheet Gradebook Template
Word Processor Marker Macro

June 1990

June AppleWorks Forum
Complete Electronic Index File
CheckWorks Templates
Change-A-File/Resurrection Disk
Alarm Clock Macro
Spreadsheet Gradebook Template
2 Macros that Call QuickSpell

July 1990

July AppleWorks Forum
6 Unpublished Articles
Frequency Distribution Macros
Spreadsheet Gradebook Template
MHM Data Base

August 1990

August AppleWorks Forum
AppleWorks 3.0 Patch Disk v. 1.5
BBS Help Files
Keith Johnson Macros
Shrink-It 3.03 Program
Spreadsheet Gradebook Template
Screen Shot Macro

September 1990

September AppleWorks Forum
Change-A-File/Resurrection Disk
CheckWorks v. 1.2
BusinessWorks Demonstration Disk
Printer Code Menu Macro

October 1990

October AppleWorks Forum
6 Unpublished Articles
Spreadsheet Gradebook Templates
Macros that Print Two-Sided Documents

November 1990

November AppleWorks Forum
Hard Drive Test Disk
Apple IIc/IIe ProDOS System v. 3.2
Personal Loan Template
Compound Interest Template
@IRR Business Templates
Page Break, SmartSave, and Number Cells Macros

December 1990

December AppleWorks Forum
Complete Index for AppleWorks Forum
Change-A-File/Resurrection Update
Info Files Macros
MacroMania
PowerMacros
Car Loan & College Savings Templates
Sample ReportWriter Reports
RamFactor Patches
Menu Macro to Manage Subdirectories

January 1991

January AppleWorks Forum
Loan Cost Estimation Worksheet
Church Membership Template
Computer Dictionary
Simplified Accounting Templates
Clemsha's TO. Modules
Keypad Remapping Macro
Number Records Program

*NAUG on Disk is only available on 3.5-inch disks.

Order Form

Mail to: National AppleWorks Users Group, Box 87453, Canton, Michigan 48187, (313) 454-1115. NAUG ships via UPS. Foreign orders by credit card only. Foreign postage additional; please specify air or surface. Payment must accompany purchase orders.

□ Please upgrade my subscription.
____ months x $7.00 = $ ________

□ Please ship these back issues.
____ Jan '90
____ Feb '90
____ Mar '90
____ Apr '90
____ May'90
____ Jun '90
____ Jul '90
____ Aug '90
____ Sep '90
____ Oct '90
____ Nov '90
____ Dec '90
____ Jan '91

____ months x $8.00 = $ ________

□ Please send the 1990 collection @ $90 $________

TOTAL $ ________

NAUG I.D. # Expiration Date

□ Check □ Visa/MC □ P.O. No. ____________

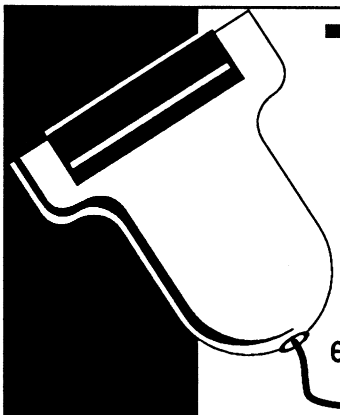
Credit Card # Expiration Date

Signature Phone Number

Name (Please print)

Address (UPS shipping - No P.O. boxes please.)

City, State, Zip



Teach your Apple II to read.

NAUG now offers InWords, a program that reads full-page documents into AppleWorks. InWords recognizes any printed typeface and has a built in text editor. Requires a Quickie scanner.

NAUG Price:	$77.95 (plus $3 s/h)
List Price:	$129.00

InWords

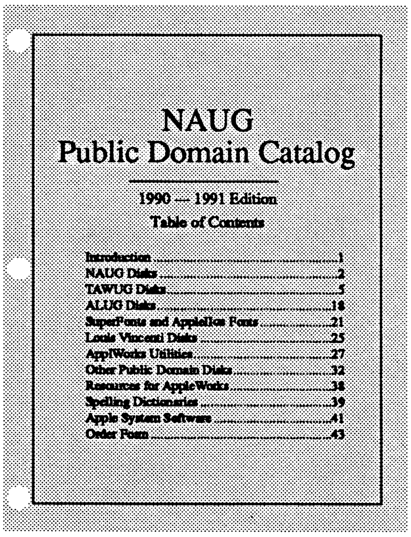
Satisfaction guaranteed or your money back.

NAUG Public Domain Catalog

NAUG's 44-page 1990-91 Public Domain Catalog lists hundreds of templates, fonts, AppleWorks enhancements and utilities that are available from NAUG. These files can dramatically increase your productivity with AppleWorks.

The catalog costs $5 and includes a rebate coupon worth $2, valid on your first order of public domain disks.

NAUG Policy: Satisfaction guaranteed or your money back.

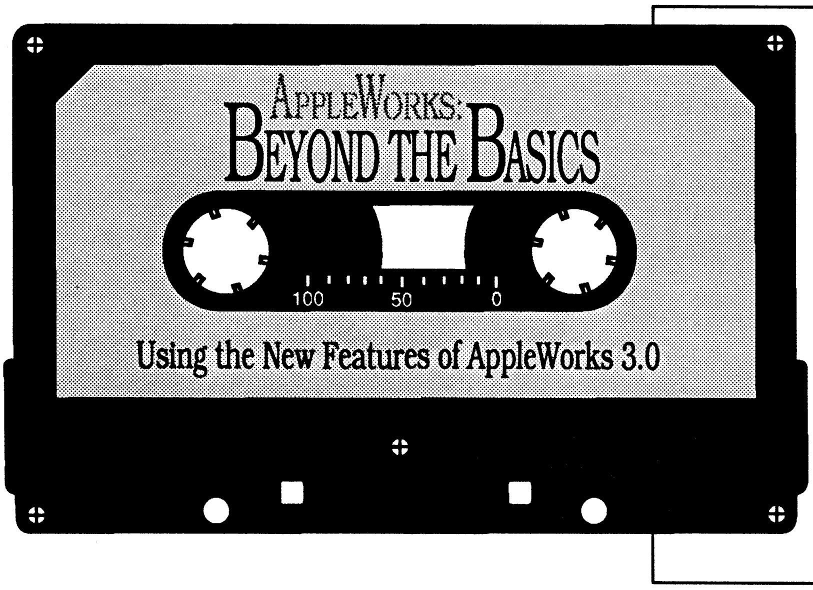


NAUG
Public Domain Catalog

1990 — 1991 Edition

Table of Contents

Introduction	1
NAUG Disks	2
TAWUG Disks	5
ALUG Disks	18
SuperFonts and AppleIIgs Fonts	21
Louis Vincenti Disks	25
ApplWorks Utilities	27
Other Public Domain Disks	32
Resources for AppleWorks	38
Spelling Dictionaries	39
Apple System Software	41
Order Form	43



Here is the newest addition to NAUG's popular series of cassette-based AppleWorks lessons. Dr. Warren Williams, the nation's leading AppleWorks instructor, presents a ninety minute introduction to the new features of AppleWorks 3.0.

This tape describes each new feature of AppleWorks 3.0 and gives a practical example of its use.

Satisfaction guaranteed or your money back.

InWords

Send me _____ copies of InWords @ $77.95 ______

Shipping @ $3.00 ______

Total ______

□ Check □ Visa/MC □ P.O. No. __________

Credit Card # Expiration Date

Signature

NAUG I.D. Number Phone Number

Name (Please print.)

Address (UPS shipping - No P.O. boxes, please.)

City, State, Zip

National AppleWorks Users Group
Box 87453; Canton, Michigan 48187; (313) 454-1115.

NAUG ships via UPS. Foreign orders by credit card only. Foreign postage additional; please specify air or surface. Payment must accompany purchase orders.

Send me ____ Public Domain Catalogs @ $5. $ ______

(Outside North America, add $3 per catalog) $ ______

Total $ ______

□ Check □ Visa/MC □ P.O. No. __________

Credit Card # Exp. Date

Signature

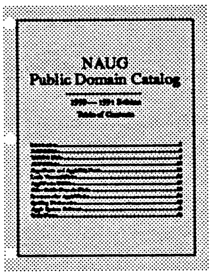
Phone

Name (Please print.)

Address

City, State, Zip

Send completed card to:
National AppleWorks Users Group
Box 87453
Canton, Michigan 48187
(313) 454-1115.



ORDER FORM

Qty.			NAUG Member	Non-Member	Total
____	AppleWorks 3.0	@	$8.95	$9.95	$ ______
____	Word Processor Tips I	@	$8.95	$9.95	$ ______
____	Data Base Tips		8.95	9.95	______
____	Spreadsheet Tips		8.95	9.95	______
____	Word Processor Tips II		8.95	9.95	______
____	Set of Five Tapes		40.00	45.00	______
				Total Amount	$ ______

Mail to: National AppleWorks Users Group; Box 87453; Canton, MI 48187; (313) 454-1115

□ Check □ VISA/MC □ P.O. No. __________

Credit Card # Exp. Date

Signature Phone Number

NAUG I.D. Number

Prices are in U.S. dollars. Payment must accompany all purchase orders.
Foreign orders by credit card only, shipping is additional. Specify air or surface.

Mail to: (Print clearly; this is your return label.)

Name

Address (No P.O. boxes, please)

City, State, Zip

How to Keep Your Financial Records in AppleWorks – Part 1

by Diana Crawford Diamond

Although most Americans associate the month of April with thoughts of federal and state income taxes, January is the time to plan the best way to keep your tax records. This month, I will describe how to use AppleWorks' data base module to keep track of the checks you write and of your tax-related expenses. Next month I will describe how to use those data to lighten your burden at income tax time. By the end of these two articles you will be able to maintain your personal financial data with AppleWorks.

You might be surprised that I use AppleWorks' data base module for these operations; many users naturally gravitate to the spreadsheet for any operation that involves numbers and money. However, the data base module lets you enter data easily in random order and can generate clear, well organized reports. The data base module also makes it easy to sort your records, find the records you want, and perform the simple arithmetic operations you need.

There are many ways to use AppleWorks to maintain your checkbook and tax records. Some systems are simple; others are more complex. The approach I will describe emulates the system used by most banks. That is, I treat each month's transactions as a separate set of activities with its own beginning and ending balance. That makes it easy to balance your checkbook and identify errors in your records. The disadvantage is that this system requires you to enter more data and generates larger files than other approaches to checkbook management.

Getting Started

Start by creating a new data base file with the categories that appear in *Figure 1.* (Do not enter the numbers 1-30; I put them here to make it easier for you to create the categories.) Accountants may prefer to call categories 5 and 6 "Debit" and "Credit" respectively, which is more accurate but less self-evident to accounting novices.

Figure 1: Data Base Categories

```
 1 Bank
 2 Date
 3 Number
 4 Payee/Source
 5 Check Amount
 6 Deposit Amount
 7 For
 8 1Item
 9 1Amount
10 2Item
11 2Amount
12 3Item
13 3Amount
14 4Item
15 4Amount
16 Statement Date
17 Tax Category
18 Code
19 Signed
20 Cleared
21 Invoice/Account#
22 Note
23 Addressee
24 Place
25 Street
26 City
27 State
28 Zip
29 Phone
30 Count
```

Your non-tax related transactions generally use categories 1-7; these categories contain the data you normally enter into your checkbook. When entering data for groceries and other non-tax related expenses, fill out categories 1-7 and then press the Apple-Down-Arrow Key combination to go to the next record.

You use categories 16-22 when you want to enter additional tax-related information about a check. Until you establish consistent tax group names, you can enter anything in the tax category, even "y" for "yes". Later you can use the Apple-R command to review all the records with non-blank tax categories and refine your entries.

The Code category further refines the organization of your tax records. I will describe that category in next month's article.

You are more likely to use categories 1-7 and 16-22 than the other categories on the screen. I suggest that you use the Apple-L command and tell AppleWorks to jump the cursor from left to right and top to bottom on the single record layout screen.

Many users have more than one checking account, and this system accommodates multiple accounts. When you enter a record, type an abbreviation in the Bank category for each different account (e.g., "B" for "Bank", "M" for "Money Market Fund".) Later you will use the Apple-R command to select records for the different accounts so you can balance your checkbooks. Create a "Bank" category even if you have only one account, then use that category to differentiate between check, cash, and credit card transactions.

Categories 8-15 let you manage multiple-source deposits. You can list up to four different sources for each deposit; enter a brief description of each deposit in the appropriate "item" category. You can also use these categories to describe items bought by check or credit card; particularly if you itemize your tax deductions.

The Statement Date entry in category 16 can help you balance your checkbook, remind you of recurring monthly expenses, and locate missing checks. Each transaction record will include the month and year you expect the check to appear on your bank statement. Then you insert the actual statement date after the check clears.

Recurring Expenses

You can also use the Statement Date category to remind you to include any automatic checkbook deductions such as mortgage payments, loan payments, and bank service charges from your bank balance.

At the beginning of the year, enter a record with the name of the payee and the amount of the deduction. Then make 11 copies of the record and enter the date the deduction is made from your account and the month and year the deduction will appear on your statement. For example, your first monthly mortgage payment might be deducted on January 15, 1991 and would appear on your February 1991 statement. These transactions will automatically appear when you balance that month's bank statement.

Optional Categories

The address data in categories 23-28 is useful if you must write or call one of your creditors or clients. I suggest that you enter the names and addresses in the appropriate categories instead of into a separate address file. You can also use these entries to generate labels, envelopes, or checks you mail in window envelopes.

Remember to always create 30 categories for your data base file. Although AppleWorks lets you add categories later, that will destroy all your report and screen formats. By creating 30 categories, you can rename unused categories later and preserve your reports and formats.

Entering Information

Your next step is to use the Apple-V command to enter some standard values. Enter the name of your most frequently used bank in the Bank category, put the month and year you expect to receive your next statement in the Statement Date category, enter the initials of the person who writes the most checks in the Signed category, and type the number "1" in the Count category. AppleWorks will automatically enter these defaults when you insert a new record in your file.

Issue an Apple-I command and enter a record that contains the current balance from each of your checkbooks. Enter "A Begin Balance" in the Payee/Source category to keep the record at the top of every alphabetical list you print. You will correct this balance later in this process.

Now enter the data for all the deposits and checks you wrote since your last bank statement.

Finding Your Current Balance

Next, you will create a report that will help you find your correct current balance in each account. Create a report called "Bank1 balance" that matches the example in *Figure 2*. Set the characters per inch to 17.

With the Report Format on your screen, follow these steps to create a calculated category that will compute your current balance:

1. Put the cursor on the characters to the right of the "Check Amount" category and issue an Apple-K command. Name the category "Totals".
2. Enter the calculation rules as "F-G". That will display positive numbers for all deposits and

Other Ways to Maintain Your Financial Data

There are at least three different sets of AppleWorks templates and programs that can help you maintain your financial data.

Diana Diamond, author of this article, offers Checkbook Plus, a collection of four well-designed templates and more than two dozen reports that can help you write checks, balance your checkbook, and maintain your tax records. Checkbook Plus is a simple system that does not use macros to link together files or commands. The Checkbook Plus disk includes a brief mini-manual and sample data files to help you get started. The templates are compatible with all versions of AppleWorks and do not require TimeOut UltraMacros.

Checkbook Plus is shareware. You get the disk from the NAUG Public Domain Library (the 5.25-inch disk costs $4, the 3.5-inch disk costs $6, plus $2 s/h per order) and send the author a $10 shareware fee if you use the templates on the disk.

CheckWorks is a collection of macros and data base files that automate the process of paying your bills and maintaining your financial records. The program provides a menu-driven environment that makes it easy to write checks and store your expense records in AppleWorks.

CheckWorks requires AppleWorks 3.0 and UltraMacros 3.1. The program is "fairware"; you get the disk from the NAUG Library and then send the author any payment you think is equitable after testing the program.

AlphaCheck and AlphaCheck Plus are commercial-quality automated check writing and financial record keeping programs that work within AppleWorks. Both products present the user with a blank check on the screen. You fill out the check and the programs use that information to print checks and maintain your financial records.

AlphaCheck is a single entry system that is suitable for most home applications. A favorable review of AlphaCheck appeared in the October 1989 issue of the ***AppleWorks Forum***.

AlphaCheck Plus adds features such as double entry bookkeeping and enhanced reporting, including trial balances, expense reports, vendor reports, cash disbursement journal reports, and, through an optional payroll module, payroll reporting.

AlphaCheck and AlphaCheck Plus require AppleWorks 3.0. The programs are compatible with, but do not require, TimeOut UltraMacros. (Both programs include a run-time version of UltraMacros on the disk.)

Both AlphaCheck and AlphaCheck Plus are available at significant discounts to NAUG members. See page 17 of the October 1990 issue of the ***AppleWorks Forum*** for the special NAUG member prices.

[ActaSoft, 19700 Wells Drive, Woodland Hills, CA 91364; (818) 996-6731.]

— Cathleen Merritt

negative numbers for all checks. Set the format for two decimal places; leave one blank space after the column.

3. Issue an Apple-T command to tell AppleWorks to total this category. A double line will appear under the column.
4. Put the cursor on the Deposits column and issue an Apple-J command to right justify this column. Indicate that you want two decimal places with one blank space after the column.
5. Put the cursor on the Check column and follow the procedures in step #4 to right justify the data in that column.
6. Issue an Apple-R command and enter the following record selection rules:

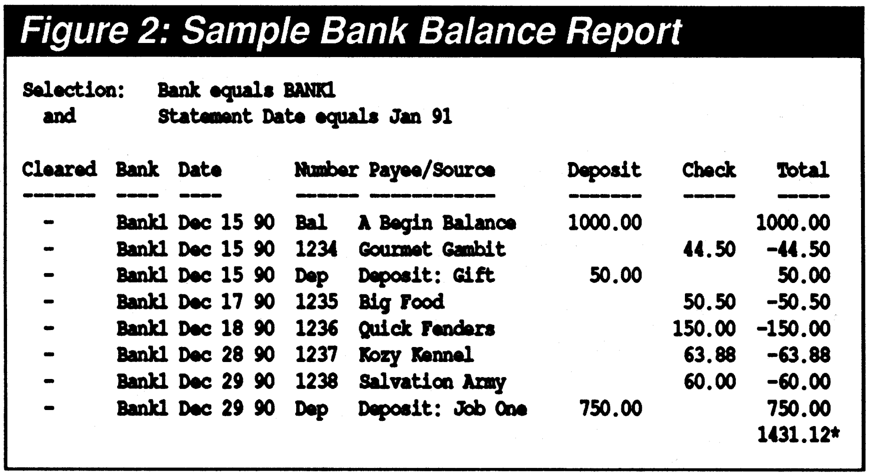
   Bank equals Bank1 (or whatever you called your first bank account) and Statement Date equals Jan 91 (or the next anticipated statement month and year).
7. Issue an Apple-P command and "print" your report to the screen. Your output should look like the example in *Figure 2*.

Your report will list each entry and present a total that represents your current balance. Use the Space Bar to scroll through a long list of entries. If "####" appears where you expect to see a number, return to the Report Format Menu, put the cursor in the Deposit, Check, or Total column, and use the Apple-Right-Arrow Keys to create additional space to accommodate larger numbers.

If "####" still appears, you may have entered something in the amounts categories that cannot be added. For example, you might have entered the letters "o" or "l" for the numbers "Ø" or "1". Return to Review/Add/Change Mode and use the Apple-R command to search for records that contain the letters "o" or "l" in the Check Amount or Deposit Amount categories. Correct your entries and regenerate the bank balance report.

If you have several bank accounts, you can copy the bank report format while at the Report Menu (select #4, "Duplicate an existing format") and

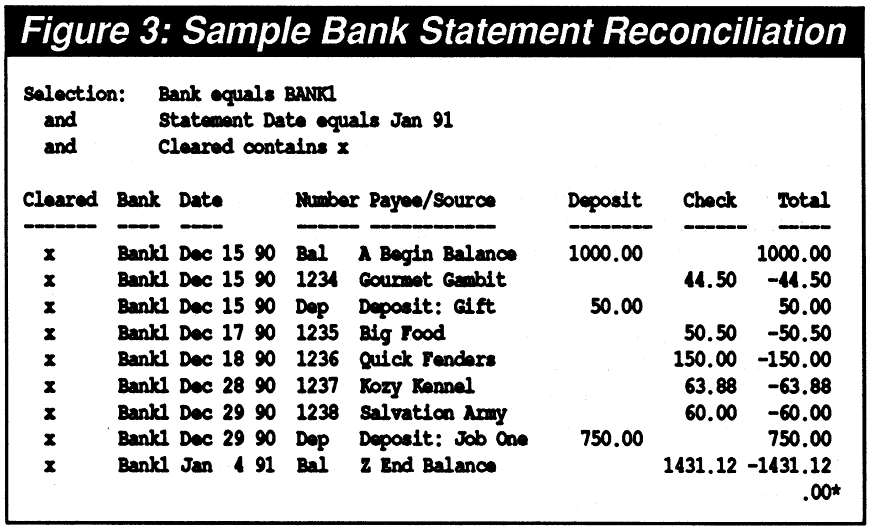
Figure 2: Sample Bank Balance Report



```
Selection:   Bank equals BANK1
  and        Statement Date equals Jan 91

Cleared Bank  Date       Number Payee/Source     Deposit    Check    Total
------- ----  ----       ------ ------------     -------    -----    -----
   -    Bank1 Dec 15 90  Bal    A Begin Balance  1000.00            1000.00
   -    Bank1 Dec 15 90  1234   Gourmet Gambit               44.50   -44.50
   -    Bank1 Dec 15 90  Dep    Deposit: Gift      50.00              50.00
   -    Bank1 Dec 17 90  1235   Big Food                     50.50   -50.50
   -    Bank1 Dec 18 90  1236   Quick Fenders               150.00  -150.00
   -    Bank1 Dec 28 90  1237   Kozy Kennel                  63.88   -63.88
   -    Bank1 Dec 29 90  1238   Salvation Army               60.00   -60.00
   -    Bank1 Dec 29 90  Dep    Deposit: Job One  750.00             750.00
                                                                    1431.12*
```

Figure 3: Sample Bank Statement Reconciliation



```
Selection:   Bank equals BANK1
  and        Statement Date equals Jan 91
  and        Cleared contains x

Cleared Bank  Date       Number Payee/Source     Deposit    Check    Total
------- ----  ----       ------ ------------     -------    -----    -----
   x    Bank1 Dec 15 90  Bal    A Begin Balance  1000.00            1000.00
   x    Bank1 Dec 15 90  1234   Gourmet Gambit               44.50   -44.50
   x    Bank1 Dec 15 90  Dep    Deposit: Gift      50.00              50.00
   x    Bank1 Dec 17 90  1235   Big Food                     50.50   -50.50
   x    Bank1 Dec 18 90  1236   Quick Fenders               150.00  -150.00
   x    Bank1 Dec 28 90  1237   Kozy Kennel                  63.88   -63.88
   x    Bank1 Dec 29 90  1238   Salvation Army               60.00   -60.00
   x    Bank1 Dec 29 90  Dep    Deposit: Job One  750.00             750.00
   x    Bank1 Jan  4 91  Bal    Z End Balance              1431.12 -1431.12
                                                                       .00*
```

change the selection rules to choose the records for each different account.

Reconciling your Checkbook and Statement

Follow these steps to generate the report that appears in *Figure 3* and to reconcile your monthly bank statement:

1. Use the Apple-R command to select all check and deposit records for the bank account that have a statement date equal to the current month.
2. Display the records in multiple record layout mode and enter an "x" in the Cleared category for all transactions that appear on your bank statement.
3. Add records that contain any interest received and fees paid. Mark these transactions cleared.
4. Create a new record and enter the bank's final balance in the "Check Amount" category by typing"Z End Balance" in the Payee/Source category.
5. Issue an Apple-P command and generate the Bank1 Balance Report you created earlier.

The sum at the bottom of the Totals column should read ".00*". Any amount other than zero tells you that your records do not match the bank's; it is time to check your data for discrepancies.

Preparing for the New Month

Now you should prepare your account for next month's statement. Proceed as follows:

1. Arrange the records in A-Z order based on the "Cleared" category. That will put all the uncleared records at the top of the list.
2. Go to the top of the list and change the Statement Date on the first transaction to the following month. Use the Apple-" command to copy this date into the Statement Date on all the uncleared checks.
3. Change the Statement Date on all cleared checks to the actual statement date.
4. Issue an Apple-V command and set the standard value for the Statement Date to the following month.
5. Insert a new record that contains the beginning balance for the month.

Conclusion

You now have an AppleWorks data base file you can use to store your tax and checkbook data and help you reconcile your monthly bank statements. Next month I will describe how to use these data to help you complete your income tax forms.

[Diana Crawford Diamond is a freelance writer and photographer living in Washington, D.C.]

CLARIS®

Last chance for this special NAUG offer

Get the newly revised *AppleWorks Resource Guide for Teachers and Parents* at *25% off* the regular price!

Now there's one easy-to-use source for ideas on how to use the AppleWorks family of products from Claris to teach math, science, social studies, and language arts to elementary and secondary school students. ***The AppleWorks Resource Guide for Teachers and Parents*** provides descriptions of the many resources — templates, books, and add-on products — that convert AppleWorks and AppleWorks GS into superb teaching tools.

The Resource Guide lists over 140 available templates and illustrates how you can develop templates into full lesson plans. The advice and suggestions provided will help you write your own templates as well as assist you in evaluating the templates mentioned in the Guide. In addition, a comprehensive supplier index makes it easy for you to order any of the materials listed.

AppleWorks Resource Guide Order Form

To send for your copy, simply mail this completed form to:
Claris Software
P.O. Box 526
Santa Clara, CA 95052
Orders must be postmarked by January 31, 1991. Allow 3 to 4 weeks for delivery. Offer valid in the U.S. only. Verbal purchases orders not accepted. Shipment against Purchase orders is subject to credit approval.

	Qty	Price (reg. $20)	Amount
AppleWorks Resource Guide		$15.00 each	
Add $3.00 per Guide for shipping & handling			
Add sales tax based on location shipped to			
Total			

Ship to:

Name

Institution (if applicable)

Address (Claris can not ship to P.O. Box)

City State Zip

Payment Information:

□ Purchase order (must be accompanied by this form.)

□ Enclosed is a check or money order for $ __________.
Make check payable to Claris Corporation.

□ Please charge the proper amount to my credit card.
Check one: □ VISA □ MASTERCARD

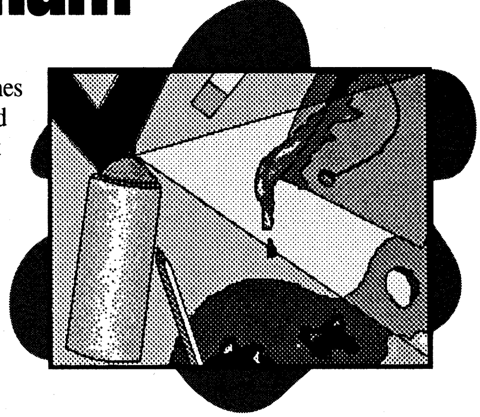
Card Number

Expiration Date

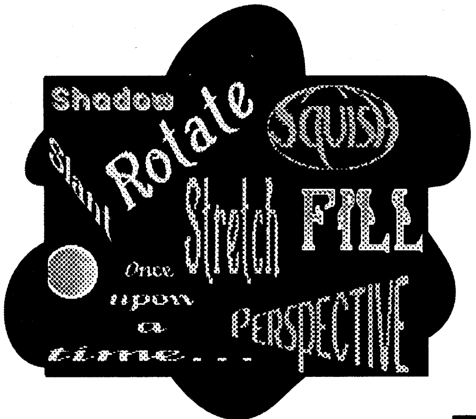
____
Name on credit card (please print)

____
Signature

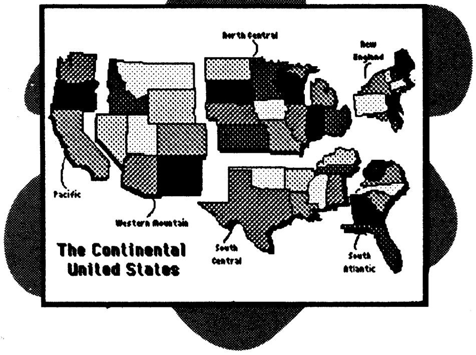
Forget the Gold... Go for the Platinum



Platinum Paint™ finishes first when it comes to paint programs. With more features and speed than Paintworks Gold, DeluxePaint II and others, Platinum Paint wins top honors in every event. The palette editor lets you quickly find the right color using any 16 of over 4,000 available. Options include controls that lighten, darken, warm, cool, mix and blend.



Compose stunning graphic and text effects without any hassle. You can use over 20 different effects with brushes, marquee and lasso selections, such as Lighten, Darken, Stretch, Slant, Resize, Rotate, 2-D Perspective, Shadow, Flip, Mirror, Invert and more.



With Platinum Paint you can easily create visual aids and student handouts. With over 100 keyboard shortcuts available, Platinum Paint lets you quickly produce graphics, like this map, that could take forever in other paint programs. All commands are easily accessible through pull-down menus and dialog boxes.

- Advanced lasso selection techniques.
- Repeat selected item as a brush stroke.
- Nine brush modes with Smear, Slide, Charcoal, Shade, Wash and Smooth.
- Six airbrush methods with adjustable spray speed and nozzle size.
- "Squish Brush" fill method contorts the brush to fit *any* shape—even circles.
- Super-fast smoothing, shading and blending.
- Fast, automatic color remapping.
- Automatic shadowing.
- Three kinds of gradient fills, including sunburst radial fill.
- Bézier curves.
- Easily colorize scanned images.
- Advanced color and area masking.
- Open four documents at once.
- Cycle colors for animation.
- Import and export all popular graphic formats.
- Compatible with System Software 5.0.3 and later (included).

Offer Extended to January 31, 1991!

Trade Up to Platinum Paint for Only $49.95

This advanced IIGS paint program improves on features from Paintworks Gold, DeluxePaint II and PixelPaint. Dozens of advanced color graphic editing features and special effects are all easy to use and incredibly fast. And as always, our unsurpassed customer support stands behind it.

If you're tired of your slow, featureless, hard-to-use, or unsupported paint program, now's the time to go for Platinum Paint. Send us the original manual cover from *any* commercial Apple II paint program and we'll take $50 off our retail price of $99.95.

With all the assets described here, plus compatibility with System 5.0.3, there's no reason not to upgrade. Mail us your manual cover before December 31, 1990, and we'll send you Platinum Paint for only $49.95 plus $5.00 shipping and handling.

$49.95 upgrade offer available only when ordered directly from Beagle Bros. Platinum Paint requires an Apple IIGS with at least 1 Mb of memory, a 3.5 inch disk drive and a color monitor. Platinum Paint is a trademark of Beagle Bros, Inc., Apple IIGS is a trademark of Apple Computer, Inc., Paintworks Gold is a trademark of Mediagenic/Activision, DeluxePaint II is a trademark of Electronic Arts and PixelPaint is a trademark of SuperMac Software.

Beagle Bros, Inc.
6215 Ferris Square, Suite 100
San Diego, CA 92121
(619) 452-5500

Data Base Tips

How to Use BASIC to Create Numbered Records

by David Jones

*Last month's **AppleWorks Forum** described how to use AppleWorks' spreadsheet module to prepare numbered data base records. This month's article describes how to use a BASIC program to accomplish that goal. We hope this article will encourage advanced AppleWorks users to develop other ways to use BASIC and other languages to enhance AppleWorks.*

If you keep track of checks, invoices, or other numbered documents, you need to create numbered data base records in AppleWorks. This article describes how to prepare a BASIC program that creates those records and how to import those records into AppleWorks.

Preparing the Program

Follow these steps to prepare a BASIC program that creates numbered records in a text (ASCII) file:

1. Type the following BASIC program into an AppleWorks word processor file:

```
10 HOME : D$=CHR$(4)
20 PRINT:PRINT:PRINT "Enter beginning record number "
25 PRINT
30 INPUT "==> ";B
40 PRINT:PRINT:PRINT "Enter ending record number "
45 PRINT
50 INPUT "==> ";E
55 PRINT D$;"DELETE RECORDS"
60 PRINT D$;"OPEN RECORDS"
80 PRINT D$;"WRITE RECORDS"
82 FOR X=B TO E
84 PRINT X
90 NEXT
100 PRINT D$; "CLOSE RECORDS"
110 END
```

2. Check your work carefully for typographical errors.
3. Issue an Apple-P command and "print" the file as an ASCII text file on a disk. If you named your disk "DATA", enter the pathname /DATA/PROGRAM.
4. Quit AppleWorks and launch AppleSoft BASIC. Get the BASIC prompt on the screen. *[Ed: See page 17 of last month's issue of the **AppleWorks Forum** if you do not know how to launch BASIC.]*
5. Enter the command "PREFIX /DATA" (substitute the name you assigned to your disk for "DATA" in this example).

6 Enter the command "-PROGRAM". BASIC will convert the program from a text file into AppleSoft BASIC. *[Ed: The hyphen before "PROGRAM" is the AppleSoft command to run or execute a program.]*

7. Enter the command "SAVE NUMBER". That will save the file as an AppleSoft BASIC program called "NUMBER".

Using the Program

Now follow these steps to use the NUMBER program to produce numbered records:

1. Issue the command "PREFIX /DATA" from within BASIC.
2. The NUMBER program assumes you have a file called RECORDS on your disk. If you have that file on the current disk, skip to step #3. Otherwise, type "LOAD NUMBERS". Then enter "55" and press the Return Key. That deletes the line that looks for the RECORDS file. Then type "RUN" and skip to step #4.

3. Type the command "-NUMBERS". This will create a text file on your disk with the record numbers you specify.
4. Launch AppleWorks and indicate that you want to create a new data base file from a text (ASCII) file. Select "RECORDS" from the list of files in the catalog and press the Return Key to accept "Tabs between categories, Returns between records". AppleWorks will create a new data base file that contains the numbered records. The number will appear in the first category of each record.

Now you will copy these records into your data base file. Continue as follows:

5. Get into multiple record layout and copy all the records to the AppleWorks clipboard.
6. Issue an Apple-Q command and switch to your working data base file. Issue an Apple-C command and copy the numbered records from the clipboard into that file. This will add records that have the numbers you specified in the first category. All the other categories will be blank.

This procedure assumes that you want the numbered record to appear in the first category. If you want the numbers to appear in a different category, use the Apple-N command to insert the appropriate number of blank categories at the beginning of the file before copying the records to the clipboard.

Conclusion

It takes a few minutes to get started with the procedures I described; you must type the record numbering program and convert the program into BASIC. From then on it takes only moments to use the program to create numbered records and transfer those records into your data base file. ▒

[David Jones is Vice President of the Special Products Division of A. H. Robins Co. in Richmond, Virginia. He writes regularly for TART, the Richmond Apple II users group.]

Using UltraMacros' Cell Command

by Steve Beville

Even if you don't write macros, I think you will find useful applications for two under-utilized UltraMacros keyboard commands.

The Cell Command (Open-Apple-Hyphen) copies the contents of the currently highlighted spreadsheet cell, current entry in a data base record, or current line in a word processor document into variable $Ø. You can then use the Solid-Apple-Ø keystroke to type the contents of variable $Ø at the current cursor position. This combination of keystrokes makes it easy to move data from any cell or data base entry to a new location.

For example, proceed as follows if you entered data in the wrong category in a data base file:

1. Put the cursor in the category that contains the data you want to move and press Open-Apple-Hyphen.
2. Move to the category where the data should go, press Open-Apple-Y to clear the incorrect data and then Solid-Apple-Ø. Finally, press the Return Key.

You can also use Open-Apple-Hyphen if you discover that you are entering data into the wrong spreadsheet cell or data base category. Don't press the Escape or Return Key. Instead, proceed as follows:

1. Press Open-Apple-Hyphen to store the information in variable $Ø.
2. Press the Escape Key to restore the cell or category to its previous condition.
3. Move to the correct category and press Solid-Apple-Ø. Complete the entry and press the Return Key. ▒

[Steve Beville is an AppleWorks and UltraMacros consultant from Spartanburg, South Carolina. Some of Mr. Beville's macros appear on Beagle Bros MacroEase disk.]

Introduction to Functions

by Dan Verkade

This is the seventh in a series of articles that describe how to use TimeOut ReportWriter to enhance the power of AppleWorks. This month's article describes how to use text functions in a report. The author assumes that you read the previous articles in this series.

You can now use ReportWriter to prepare relational reports that include calculated fields. This month, you will learn how to use ReportWriter's built-in functions to enhance your reports.

Function Basics

"Functions" are pre-programmed operations that are difficult or impossible to produce with mathematical operators. For example, it is impossible to use ReportWriter's built-in mathematical operations of addition, subtraction, multiplication, and division to determine the square root of a number. Therefore, ReportWriter offers a "square root function" (@SQRT) which can compute the square root of a number.

The syntax of the square root function is:

```
@SQRT(Field)
```

where "Field" contains a number or a reference to a ReportWriter field that contains a number. When you enter @SQRT(Field) in a report definition, ReportWriter will print the square root of the number in the Field category for each record.

The rules for using functions in ReportWriter parallel those used in the AppleWorks spreadsheet. Specifically, functions follow this syntax:

```
@FUNCTION(arguments)
```

where FUNCTION is the name of the function and the arguments consist of one or more field names, numbers, strings, or another function. Some functions require more than one argument separated by commas.

Each function must receive the correct number and type of arguments. If a function yields results that differ from what you expect, you should first check the arguments within the function.

Nested Functions

You can "nest" an unlimited number of functions within other functions as long as the formula does not exceed a maximum length of 127 characters. For example, @SQRT(@SQRT(@SQRT(256))) is a nested function. It says, print the square root of the square root of the square root of 256. ReportWriter would print the number two. (You can confirm that two is the correct result by entering 256 into a calculator and pressing the square root key three times.)

Nested functions operate from the inside out. That is, the inner-most functions operate first; the outer-most functions work last. Thus, you must read nested function formulas from the inside out. Start by determining what the inner-most function does and use the result of that calculation to determine the operation of the next outer function, and so forth.

Text Functions

Text functions manipulate and evaluate strings of text. Some require text arguments, some require numeric arguments, and some require both. The text can be a constant (i.e., text that you build into the formula), the name of a field containing text, or another function returning text.

Here are descriptions of ReportWriter's text functions; I will give examples of their use in a tutorial that follows these descriptions.

@LEN(String): Returns the length of the string within the parentheses. The @LEN function returns a numeric result. To determine the length of the word AppleWorks, enter the formula @LEN("AppleWorks"); the number 10 will appear in the report.

If you have a file with the field "Word" and four records containing the words National, AppleWorks, Users, and Group in the Word field, you could set up the formula @LEN(Word). ReportWriter would return the numbers 8, 10, 5, and 5.

@CAPFRST(String): Capitalizes the first letter of each word in the string of text. If you have book titles in the Title field in a series of records, "The Life and Times of Abraham Lincoln" becomes "The Life And Times Of Abraham Lincoln" after being manipulated by @CAPFRST(Title).

@UPPER(String): Changes every letter in the string to upper case. If Title contained "The Life and Times of Abraham Lincoln", @UPPER(Title) would return "THE LIFE AND TIMES OF ABRAHAM LINCOLN".

@LOWER(String): Changes every letter in the string to lower case.

@LEFT(String,Number): Returns the characters specified at the beginning of the string. For example, @LEFT("Karl Verkade",4) returns "Karl". That lets you get a subset from the beginning of a string and truncate a string.

@RIGHT(String,Number): Returns the specified number of characters from the end of the string. For example, @RIGHT("Heidi Verkade",7) returns "Verkade". There are numerous applications of @RIGHT. For example, you can use @RIGHT to capture the last three characters from an alphanumeric catalog number.

@MIDDLE(String,Number1,Number2): Captures the characters starting Number1 characters from the beginning of String. Number2 tells @MIDDLE how many characters to capture. For example, @MIDDLE("Betsy Warner Verkade",7,6) returns "Warner" (every space also counts as a character). @MIDDLE lets you capture any contiguous subset of characters from a string.

@TEXT(String): Returns the contents of String and is probably the most useless of all the functions. I had a good reason when I wrote it, but I can't remember why! @TEXT("AppleWorks") will return "AppleWorks".

@NUM2TXT(Number1,Number2): Converts a number from a numeric format to a text format. Number1 is the number to convert. Number2 is the number of decimal places to capture. The @NUM2TXT function rounds Number1. If you enter Ø for Number2, ReportWriter captures as many decimal places as necessary (up to 7).

Examples: @NUM2TXT(1.226,2) returns 1.23. @NUM2TXT(1.226,4) returns 1.2260. @NUM2TXT(1.226,0) returns 1.226.

@NUM2TXT lets you include numeric amounts in text messages such as "Please remit your balance of $128.25 before December 15".

@CONCAT(String1, String2, StringN): Builds strings from sets of smaller strings. (Programming languages call this process "string concatenation".) For example, you can use @CONCAT to combine data from three distinct City, State, and ZipCode fields into a single line. In AppleWorks data base labels reports you use the left justify command. With ReportWriter, you enter @CONCAT(City," ",State," ",ZipCode).

@CONCAT gives you more flexibility than AppleWorks' Justify Command. For example, @CONCAT makes it easy to separate the state and Zip Code by two spaces. @CONCAT also lets you combine a date and an amount with a message, like this:

@CONCAT("Your balance due as of ",DueDate," is ",AmountDue," ")

Which would return the line "Your balance due as of Nov 21 90 is $54.65."

@FIND(String1,String2): Finds the occurrence of String1 inside String2 and returns the starting position of String1. It returns a zero if String1 does not occur within String2. Thus, any number other than zero indicates that String2 exists within String1.

The @FIND function is case sensitive; @FIND(BCD,ABCDEF) returns "2" and @FIND(bcd,ABCDEF) returns "Ø". However, @FIND(@UPPER(bcd),ABCDEF) returns "2".

Function and Formula Tutorial

Now let's use these functions to generate a ReportWriter report. I suggest that you work at the computer; it's easier to understand these steps if you perform the operations and study their effects.

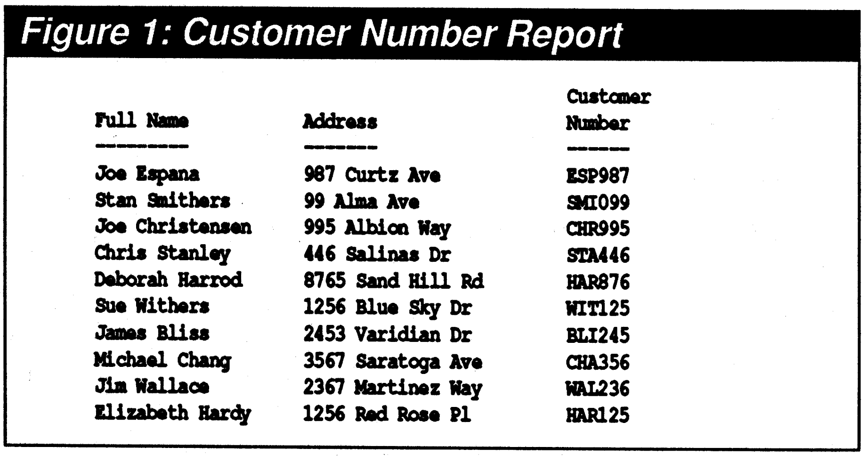
ReportWriter Tutorial...

Several months ago I described how to use unique customer codes as "keys" to form relationships between data files. Each code consisted of the first three letters of the customer's last name and the first three digits of his or her street address. We added leading zeros to any street address that did not contain three zeros.

This month you will use ReportWriter's text functions to produce these customer codes. You will need the Rolodex file from the ReportWriter disk. Your completed report will look like the example in *Figure 1.*

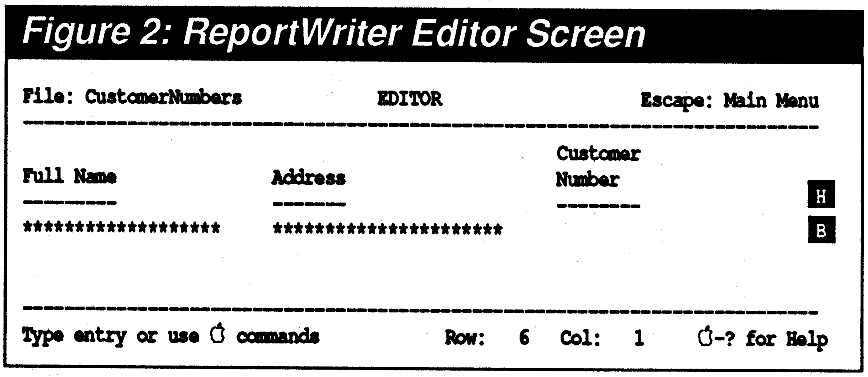
As you work, you will realize that manipulating strings of text uses a different logic than the logic you use when manipulating numbers. At first the formulas will seem daunting. However, the task will become more manageable as you get some practice.

Figure 1: Customer Number Report



```
                                              Customer
Full Name            Address                  Number
---------            -------                  ------
Joe Espana           987 Curtz Ave            ESP987
Stan Smithers        99 Alma Ave              SMI099
Joe Christensen      995 Albion Way           CHR995
Chris Stanley        446 Salinas Dr           STA446
Deborah Harrod       8765 Sand Hill Rd        HAR876
Sue Withers          1256 Blue Sky Dr         WIT125
James Bliss          2453 Varidian Dr         BLI245
Michael Chang        3567 Saratoga Ave        CHA356
Jim Wallace          2367 Martinez Way        WAL236
Elizabeth Hardy      1256 Red Rose Pl         HAR125
```

Figure 2: ReportWriter Editor Screen



```
File: CustomerNumbers          EDITOR                    Escape: Main Menu
--------------------------------------------------------------------------
                                                  Customer
Full Name              Address                    Number
---------              -------                    -------                 H
*******************    **********************                             B



--------------------------------------------------------------------------
Type entry or use  commands              Row:   6   Col:   1     -? for Help
```

Follow these steps to generate the report:

1. Add the Rolodex file to the Desktop and invoke ReportWriter from the TimeOut Menu.
2. Select choice #1, Edit a ReportWriter definition, and choice #3, Make a new file. Name your new definition file CustomerNumbers.
3. Build the layout to look like the example in *Figure 2*. The longest name in Rolodex is 19 characters; the longest address is 22 characters. Do not add the field for Customer Number yet.

You must postpone the definition of the Customer Number field because the report must perform several calculations to generate this "number". Since some of the calculations are based on the results of other calculations, the sequence of the calculations becomes important.

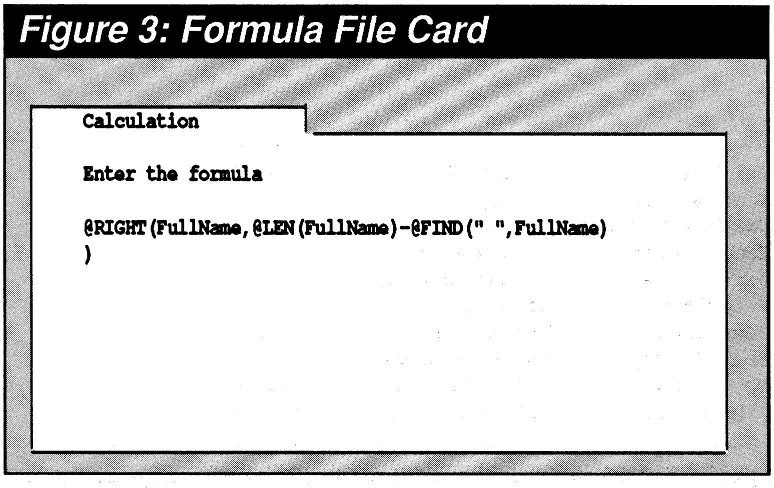
ReportWriter always performs calculations in field number order. That is, it calculates field #1 before field #2, and field #5 before field #9. ReportWriter assigns field numbers sequentially in the order in which you add field markers, not the order in which you define each field. Thus, it becomes important to add the field markers in the order I describe in this tutorial.

4. Follow these steps to define the different fields:
   A. Put the cursor on the field markers below the heading Full Name and enter an Apple-N. Assign the field name "FullName". The source is the Master File called "Rolodex". Use the category "Name" and press the Escape Key to return to the Editor.
   B. Put the cursor on the field markers under the heading Address and define this field with a field name of "Address", with a source of Master File. Use the category "Street".
   C. Set the Heading and Report Body section Position Markers as they appear in *Figure 2.*

Now you will do the necessary calculations.

5. Return to the Editor screen and put the cursor in row 6, column 1 (this is the work area below the body of the report). Use the Control-F keys to mark a new field. You will not print this field, so the length is unimportant. Issue an Apple-N command. Name the field LastName and use Calculation as the source.

Figure 3: Formula File Card



```
Calculation

Enter the formula

@RIGHT(FullName,@LEN(FullName)-@FIND(" ",FullName)
)
```

Now you will write a formula that will extract the last name of each customer from the FullName field. The formula will identify the last name by looking for the space between the first and last names.

6. Select Formula from the Define Field Menu and enter the following formula into the Formula file card:

```
@RIGHT(FullName,@LEN(FullName)-@FIND(" ",FullName))
```

Your screen should look like the example in *Figure 3*. (The file card only holds 50 characters on a line, so the 51st character jumps to the second line.)

Reading from the inside out, this formula says, "Find the length of FullName and the number of characters up to and including the space in FullName. Then subtract the number of characters up to and including the space in FullName from the total length of FullName. Finally, capture that many characters from the end of FullName."

For example, imagine that FullName contains "Mickey Mouse". The formula determines that there are seven characters up to and including the space between "Mickey" and "Mouse". It subtracts seven from the total length of the string ("Mickey Mouse" includes 12 characters, counting the space), which yields a result of five. It then takes the five right-hand most characters which consist of "Mouse". Thus, this formula returns "Mouse" for a record that contains "Mickey Mouse" in the FullName field.

7. Press the Escape Key to return to the Define Field Menu. ReportWriter will examine your formula for common errors such as missing parentheses and incorrectly spelled field names and functions. The program displays an "Inconsistent Formula" message if it finds a misspelled entry or a field name it doesn't recognize. If you get an error message, press the Space Bar to clear the message and enter an Apple-U to edit the formula.

8. The Define Field Menu will now show a "+" in the right margin of the formula to indicate that the formula is too long to fit on the display; you must choose "Formula" to see the complete formula.

9. Press the Escape Key to return to the ReportWriter Editor and create a field in the work area to the right of the LastName field. Enter an Apple-N and name this field "StreetNumber". Indicate that the field is calculated and enter the formula:

```
@LEFT(Address,@FIND(" ",Address)-1)
```

This formula starts by finding the number of characters up to and including the first space in the Address field. The "-1" tells ReportWriter to compute a value that is one less than that number; this yields the number of characters in the house number. (That is the number of characters to the left of the space.) The formula then uses that number in the formula @LEFT(Address,...) to capture the characters to the left of the space.

The formula finds the full street number and we only want the first three numbers. We must modify the formula as follows:

10. Issue an Apple Up-Arrow to put the cursor at the beginning of the formula. Then use the insert cursor to type "@LEFT(" at the beginning of the formula. Enter an Apple Down-Arrow to move to the end of the formula and type ",3)". The revised formula should look like this:

```
@LEFT(@LEFT(Address,@FIND(" ",Address)-1),3)
```

The outer @LEFT function truncates the string to 3 characters. Study the formula to see how it works.

11. Press the Escape Key twice to return to the Editor and create another field called ZeroNumber. This field will attach leading zeroes to the street

number if the street number is less than three characters long. The strategy will be as follows:

A. Use the @LEN function to determine the length of the street number.

B. Subtract that number from three to determine the number of zeros you need to add to a short number.

C. Use the @LEFT function to select that number of zeros from a set of three zeros.

D. Use the @CONCAT function to put together the number of zeros we need with the street number.

Here is the formula that does the necessary calculations:

```
@CONCAT(@LEFT("ØØØ",3-@LEN(StreetNum)),StreetNum)
```

Lets examine this formula by using an example of a person who lives at 4 Arch Drive. As always, we will start by working with the inner-most parentheses within the formula.

A. The "@LEN(StreetNum)" segment of the formula determines that there is 1 character in the StreetNum field for this record.

B. "3-@LEN(StreetNum)" subtracts 1 from 3 to yield 2. Thus, you need to add 2 zeros before street number.

C. "@LEFT("ØØØ",3-@LEN(StreetNum))" selects the first two zeros in the string ØØØ.

D. "@CONCAT" puts together those two zeros with the number 4 in StreetNum.

The result of this operation is "ØØ4", the first characters we need for the customer number.

12. Press the Escape Key twice to return to the Editor. Now you are ready to use the "computed" string in the ZeroNum field to compute a customer number.

13. Create a six-marker field underneath the heading Customer Number in the report body. Name this field CustomerNumber and use this formula:

```
@CONCAT(@UPPER(@LEFT(LastName,3)),ZeroNum)
```

The inner-most portion of this formula extracts the first three letters of the person's last name from the LastName field. The formula then forces those letters into upper case. Finally, the formula concatenates the three characters from the last name and the street number with any leading zeroes.

14. Enter an Apple-G to generate the report. You should get the results that appear in *Figure 1.*

Think of the time this report can save when you need to create customer numbers for 2,000 customers! In a future article I will describe how to put this list of customer codes directly into an AppleWorks file.

[Dan Verkade is the author of TimeOut ReportWriter, DoubleData, SuperForms, and other popular AppleWorks enhancements.]

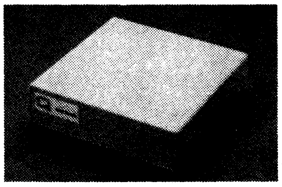
Corrections

December 1990, page 25:

The Apple IIe emulation card for the Macintosh LC computer has a suggested retail price of $199.

QUALITY COMPUTERS OVER-NIGHT!

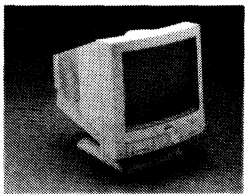
1-800-966-1508 or 1-800-443-6697



QC HARD DRIVE

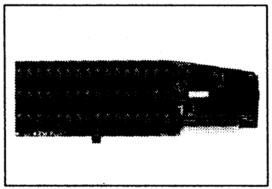
Comes with Conner mechanism, Apple High Speed SCSI card (add $100 for RamFAST card), and a plain English manual. Many features!

40 MEG.	**$599.95**
100 MEG.	**999.95**



MAGNAVOX RGB MONITOR

An affordable addition to your IIe or IIGS sytem. Features 13" diagonal screen, built-in tilt stand, 2,000 char. display, vertical and horizontal controls, analog and digital input, and more! (IIGS req. cable) **$299.95**



AE GS-RAM PLUS

Up to 8 MEG. of DMA compatible RAM at your fingertips. Features: AW expander, AW printer buffer, diagnostic software.

1 MEG.	**$219.95**
2 MEG.	**259.95**

HARD-DISKS

NEW! **QUALITY COMPUTERS (Apple SCSI)**	
40 MEG. (Conner Drive)	599.95
100 MEG. (Conner Drive)	999.95
CMS (Apple SCSI)	
40 MEG. (Conner Drive)	649.95
60 MEG.	699.95
45 MEG. (Removeable)	899.95
APPLIED ENGINEERING	
40 MEG. Vulcan (IIe, IIGS)	649.95
100 MEG. Vulcan (IIe, IIGS)	1,285.95

MEMORY CARDS

AE RAMWORKS III	
256K	139.95
512K	159.95
1 MEG.	209.95
AE RAM EXPRESS II	
256K	159.95
512K	179.95
1 MEG.	219.95
AE GS-RAM PLUS	
1 MEG.	199.95
2 MEG.	259.95
3 MEG.	319.95

MEMORY CHIPS

256K (bank)	24.95
256Kx4 (bank)	25.95
1 MEG. (bank)	79.95

DISK DRIVES

AE 3.5"	199.95
AE 5.25"	119.95
Apple 3.5" (IIc Plus, IIGS)	319.95
Apple 5.25"	249.95

PRINTER INTERFACES

AE Parallel Pro	79.95
AE Serial Pro	109.95
SPECIAL! SuperWriter 924	64.95

PRINTERS

ImageWriter II	459.95
Panasonic 1180	199.95
Panasonic 1124	329.95

MODEMS

AE DataLink Express w/MNP5	189.95
SupraModem 2400	109.95

MONITORS

Magnavox or Goldstar 12"	99.95
NEW! Magnavox RGB	299.95
Apple RGB	459.95

SOUND & GRAPHICS

Quickie Scanner	199.95
ComputerEyes IIGS	195.95
Apple Video Overlay Card	429.95
HyperStudio	86.95

COMPUTERS

Apple IIGS CPU (w/1 MEG. RAM)	825.95
Laser 128EX-2 (3.5")	449.95

ACCESSORIES

AE Conserver	77.95
AE PC Transporter	259.95
NEW! AE Power Supply IIGS	79.95
Apple IIe Enahancement Kit	59.95
Zip GS	from 189.95

BUSINESS SOFTWARE

AppleWorks 3.0	169.95
BeagleWrite	48.95
BusinessWorks Bundle	249.95
DB Master Pro	189.95
Quicken	38.95

EDUCATIONAL SOFTWARE

Talking Reader Rabbit GS	
Home	34.95
School	49.95
Lab Pack	99.95
MathBlaster Plus	32.95
AlgeBlaster Plus	32.95
Read & Roll	34.95
Math Talk GS	27.95
Mavic Beacon (IIGS)	33.95
McGee	29.95
StudyMate	27.95

GAMES

Battle Chess GS	32.95
Crystal Quest	32.95
Hunt for Red October	22.95

GRAPHICS

Design Your Own Home GS	59.95
GraphicWriter III	74.95
PrintShop NEW! (3.5 or 5.25)	29.95

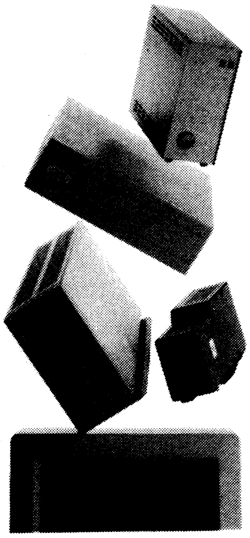
TIMEOUT SERIES

DecisionPak	89.95
StylePak	79.95
PerformancePak	69.95
MacroEase	25.95
Telecomm	42.95

UTILTIES & LANGUAGES

Apple IIGS System Software 5.0	44.95
Copy II Plus	25.95
CrossWorks	69.95
Orca/M	39.95

Hard-Disk Headquarters



Quality Computers was one of the first Apple II companies to sell and support a full line of hard-disk drives, and we're still the leader. Why? We offer more - more service, more support, more hard-disk systems, and more innovative products.

HARD-DISK BUYERS' GUIDE

Base your decision to buy a hard-disk on fact, not fiction. The Hard-Disk Buyers' Guide is packed with brand name evaluations, performance charts, and informative articles. **$6.00 (Shipped overnight to most locations)**

ASK ABOUT OVERNIGHT DELIVERY!

Quality Computers

POWER FOR PERFORMANCE

BUINESS HOURS

SALES: 1-800-966-1508 or 1-800-443-6697 Mon.-Fri. 9AM-9PM Sat. 10AM-4PM (Eastern Time)
TECHNICAL SUPPORT: (313) 774-7740 Mon.-Fri. 10AM-5:30PM (Eastern Time)
CUSTOMER SERVICE: (313) 774-7200 Mon.-Fri. 9AM-5PM (Eastern Time)

20200 E. Nine Mile Road • Box 665 • St. Clair Shores, MI 48080
Orders & P.O.'s by FAX (313) 774-2698

My Favorite Macro

Enhance the Numeric Keypad

by Keith Johnson

The availability of a numeric keypad on Apple IIGS and late model IIe computers makes it easy to enter data into AppleWorks. Unfortunately, the Apple keypads do not provide all the cursor control and function keys included on larger, special purpose keypads. For example, there is no key that lets you blank the current cell. And pressing the Enter Key does not move the cursor to the next cell.

Doak C. Cox of Honolulu, Hawaii does a significant amount of data entry, so he wrote a macro that makes it easier to use the Apple numeric keypad to enter data. *Figure 1* presents Mr. Cox's macro which changes the functions of the keys he doesn't use during the data entry task to the functions he wants. The macro remaps the function of five keys so they generate a different keystroke and sends all other keystrokes through unaltered.

Interesting Features

There are some interesting programming elements in this macro. For example, once started, the macro continues until the user presses the Escape Key. However, this is not the Escape generally used to stop a macro. This macro looks for the Escape keystroke, then stops itself. If you delete the line `<if x = 27... >` there is no way to stop the macro except by resetting the computer. Obviously, you must be careful when you write macros that accept user input.

The "*" key serves as the Escape Key when the macro is active. For example, pressing the "*" Key with the macro active either clears the current entry or takes you to the Main Menu. If you use the "*" to go to the Main Menu, you can press "*" again and return to the spreadsheet.

This can lead to some confusing situations. For example, if you press the "*" to return to the Main Menu and then press Return, AppleWorks will highlight the second choice on the menu instead of going to the Add Files Menu. Remember that the macro is still active. Since the macro remapped the Return Key to work like the Down Arrow, pressing Return simply moves the highlight to the second choice on the menu. Advanced macro writers can add error-checking routines to the macro to avoid this problem.

UltraMacros 3.x Techniques

UltraMacros 3.x offers an easier way to specify the keys you want to change. For example, UltraMacros 3.x lets you enter `<if x = #'/'...>` instead of `<if x = 47...>`. (The character "/" has the ASCII value of 47.) Either way works, but the second approach is easier to understand after you've been away from the macro for a while. However, you cannot use this method with unprintable characters like the Arrow Keys and the Return Key.

Figure 1: Keypad Remapping Macro

```
start
<ba-K>:<asp :                           { Works only in the spreadsheet environment.        }
msg " Keypad modification active. To deactivate press ESC on main keyboard " :
                                        { Display this message to alert the user.           }
 begin :                                { Start a loop.                                     }
  x = key :                             { Capture the next keystroke.                       }
  if x = 13 then x = 10 : else :        { Return Key becomes Down Arrow.                    }
  if x = 43 then x = 11 : else :        { Plus Key becomes Up Arrow.                        }
  if x = 47 then oa-B rtn : rpt : else : { Divide Key blanks the cell.                      }
  if x = 42 then esc : rpt : else :     { Multiply Key works like Escape.                   }
  if x = 27 then msg "" : stop : endif : { Escape clears the message and stops the macro.   }
 print chr$ x : rpt>!                   { Send the keystroke to AppleWorks and repeat the loop. }
```

New Disks in NAUG's Public Domain Library

NAUG

NAUG recently assumed responsibility for distributing the complete collection of Resources for AppleWorks disks. Lee Hayward, founder of Resources for AppleWorks, updated and reorganized the collection of templates and files for NAUG and released the disks to NAUG for distribution.

NAUG has also prepared a four page update to its Public Domain Library Catalog. To get the list of files on the Resources for AppleWorks disks and the Public Domain Library Catalog update, send a self-addressed, stamped #10 envelope to Public Domain Update, NAUG, Box 87453, Canton, Michigan 48187.

Church Finances Management

The NAUG Public Domain Library now includes the Church Finances Management System (CFMS). The files on this disk will help you maintain the financial records for a church, synagogue, or other non-profit, contribution-based organization.

CFMS is a complete financial management system that tracks income, expenses, contributions (which can be earmarked for up to 75 designated purposes), member identification, checks, and holding funds. The system does monthly and weekly accounting, budget planning and maintenance, operating expense fund management, finance campaign management, and church payrolls. The system generates reports to members, reconciliation reports, and donation reports. The disks include more than 50 pages of well written documentation.

The Church Finances Management System comes on two 3.5-inch disks and requires AppleWorks 3.0 running on a computer with at least 1-megabyte (and preferably 1.5-megabytes) of RAM and a 3.5-inch or hard disk drive.

The author, Lester Rollins of Sandy Lake, Pennsylvania, requests a $10 donation if you want telephone help with the system. Our thanks to Mr. Rollins for developing these valuable templates and contributing this work to the NAUG Library.

Two 3.5-inch disks; $12 plus $2 s/h *per order*.

Church Membership

NAUG's Public Domain Library also includes Lester Rollins' Church Membership Disk. Church Membership contains two spreadsheet templates that maintain your church and Sunday School membership and attendance records. The spreadsheets are self-documented and include all the necessary directions.

The files on the Church Membership Disk are compatible with all versions of AppleWorks and require a 100K AppleWorks desktop.

Computer Dictionary

The NAUG Public Domain Library now includes the Computer Dictionary Disk, an AppleWorks 3.0 and TimeOut QuickSpell-compatible custom dictionary that contains more than 1,500 words related to computers and computing. An AppleWorks data base file on the disk makes it easy to edit the list and add words to the dictionary. A text file on the disk is ready to use as a custom dictionary with AppleWorks 3.0 or QuickSpell. A word processor file on the disk contains all the necessary documentation.

DB Master Version Five

DB Master Version Five is a powerful data base manager that supports up to 200 fields per record and up to 250 characters per field. The program lets you work with large data files on a hard disk or on multiple 3.5-inch or 5.25-inch disks. DB Master offers powerful data manipulation and reporting capabilities and was chosen by inCider Magazine as one of the ten best programs of 1988.

DB Master requires an enhanced, 128K Apple IIe, an Apple IIc or IIc Plus, an Apple IIGS, or a Laser 128 computer, and two 5.25-inch disk drives, one 3.5-inch drive, or a ProDOS-compatible hard disk.

DB Master Version Five is shareware; you send the developer $45.00 if you use the program. Stone Edge Technology then sends you the latest version of the program, a full instruction manual, and a special offer for DB Master Professional, the relational data base version of the program. Technical support is available to registered users on a pay-per-call basis or for an annual fee. Prior to the shareware release, DB Master Version Five had a suggested retail price of $179.00.

DB Master Version Five comes on two dcuble-sided 5.25-inch disks ($8) or one 3.5-inch disk ($6) plus $2 s/h per order.

Fonts Disks

William Davis recently updated **NAUG**'s fonts disks; all the fonts are now compatible with AppleWorks GS, BeagleWrite, and all other 16-bit Apple IIGS programs that use standard IIGS fonts. These fonts remain compatible with TimeOut SuperFonts. The font families are unchanged on each disk; a complete list appears in the Fonts Disks section of **NAUG**'s 1990-1991 Public Domain Library Catalog.

GS/OS 5.0.3

The **NAUG** Public Domain Library now includes GS/OS 5.0.3, the latest version of Apple's 16-bit operating system for Apple IIGS computers. GS/OS 5.0.3 includes dozens of bug fixes and internal changes that enhance the system's performance over version 5.0.2. The new system includes ProDOS 8 version 1.9 and BASIC.SYSTEM version 1.4.1. Apple IIGS owners who use GS/OS should update to version 5.0.3.

NAUG also has the 21-page GS/OS 5.0.3 Release Notes. These are Apple's technical notes about the changes to GS/OS. The notes are designed for developers and programmers; most AppleWorks users will not need these notes. The GS/OS 5.0.3 Release Notes cost $5 including s/h ($7.50 for non-members) from **NAUG**. Include your **NAUG** membership number with your order.

Clemsha's TO. Modules

Clemsha's TO. Modules disk includes three TimeOut applications that add useful features to AppleWorks. TO.MOUSE makes it easy to display inverse, mousetext, and underlined text in any word processor document on your screen. A useful table on the disk depicts the relationship between all mousetext and keyboard characters.

TO.TAB converts spaces to tabs and tabs to spaces and makes it easy to insert true tabs in AppleWorks 1.x and 2.x files you import into AppleWorks 3.0. TO.TAB also lets you import data into the AppleWorks spreadsheet module from text files prepared by MS-DOS programs like Lotus 1-2-3. (Lotus 1-2-3 text files contain spaces between columns; TO.TAB converts those spaces into tabs so you can read the files into the AppleWorks spreadsheet.) TO.TAB also helps assembly language programmers use AppleWorks to generate source files.

TO.RS232 is a TimeOut module that sends any word processor document to a serial interface in slot 2. You can use TO.RS232 to transfer files at high speed between computers over direct connect cables or at regular speeds over modems.

Clemsha's TO. Modules requires AppleWorks 3.0 enhanced with TimeOut 3.01. This disk contains TimeOut applications, not macros. You do not need TimeOut UltraMacros to use the files on this disk.

Our thanks to **NAUG** member Barclay Clemsha of San Paolo, Brazil for writing these powerful TimeOut applications and for donating this work to **NAUG**'s Public Domain Library.

Simplified Accounting

NAUG's new Simplified Accounting Disk includes a set of cash receipts and cash disbursement templates prepared by **NAUG** member Irving Tessel.. Mr. Tessel developed these data base templates and reports to maintain the financial records for a public service volunteer organization. A short word processing file on the disk describes how to get started. You will need a basic understanding of accounting terminology to use the files on this disk. These templates will fit on any Apple II system and work with any version of AppleWorks.

The Simplified Accounting Disk also includes a set of automated bookkeeping macros prepared by Dr. Brian McDonnell of Brisbane, Australia. Dr. McDonnell's macros and sample data files were designed to serve as a tutorial on macro-driven bookkeeping. The macros should interest macro writers and anyone who uses macros to automate their AppleWorks bookkeeping system. Requires AppleWorks 3.0 enhanced with UltraMacros 3.1.

How to Get Disks

Unless otherwise noted, all disks are available in both 5.25-inch ($4) and 3.5-inch ($6) format, plus $2 s/h *per order*. Order from **NAUG**, Box 87453, Canton, Michigan 48187. Most **NAUG** disks are also available for downloading at no charge from **NAUG**'s electronic bulletin board (the Electronic Forum), and from the **NAUG** areas on CompuServe, America Online, and GEnie. Shareware payments go directly to the author, not to **NAUG**. ▒

Data Base Record Selection Rules

by Steve Ellis

Most AppleWorks users realize that the Apple-R command in the program's data base module lets you define a series of powerful selection rules. However, many users don't realize that there is a relationship between rules set in the program's Review/Add/Change mode and those set in the Report Format area.

Here are some guidelines to help you control the record selection process:

1. Record selection rules entered into a Report Format apply only to the specific report containing those rules.
2. Record selection rules created in the Review/Add/Change mode apply to the entire data base file.
3. Record selection rules created in the Review/Add/Change mode automatically apply to all the existing and new reports in that file that do not contain their own record selection rules.
4. Setting the record selection rules to "All" in the Report Format area overrides the record selection rules set in the Review/Add/Change area for that printing of the report. However, when you leave that specific report format, the rules for that report revert back to the rules set with Review/Add/Change. ▒

[Steve Ellis is the AppleWorks SIG leader and Beagle Buddy for the Orange Apple Computer Club in Orange County, California. He is the author of the Mail Merge macro set on the MacroTools II disk.]

We wish our members a happy, healthy, and successful New Year.

— From the editors and staff at NAUG

Members Helping Members

Apple IIGS Hardware and Software

by Nanette Luoma

Each month, the ***AppleWorks Forum*** lists the member-volunteers who offer technical support for AppleWorks products. This month's list identifies the volunteers who can answer questions about Apple IIGS hardware and software. Next month's issue will contain a list of members who offer help with TimeOut products and other AppleWorks enhancements from Beagle Bros.

How to Use this List

To the left of each volunteer's name are numbers indicating the items the consultant supports. Volunteers are listed alphabetically by state.

1 = Hardware
2 = Diversi Key
3 = SoftSwitch
4 = TransWarp GS
5 = GS/OS
6 = Desk Accessories
7 = AppleWorks GS

		City	Home	Work
California				
1,5-7	Dan Balsley	San Ramon	415-829-5085	
1,4,7	Rolf C. Freerks	San Pedro	213-833-8266	213-337-1333
Colorado				
1,5	Lyle Graff	Littleton	303-794-5970	303-977-4557
Connecticut				
7	Martin Knight	Middletown	203-346-9698	203-347-8594
Florida				
1,4-6	H.Clay Bailey III	Jacksonville	904-744-2499	904-725-3477
1,4	Robert J. Booz	Port Richey	813-868-1802	
7	Ronald Stankiewics	Patrick AFB		407-494-2227
1,4-7	Jeff Strichard	Ft. Lauderdale	305-587-9590	
1	Mike Ungerman	Oviedo	407-366-0060	407-366-0156
Indiana				
1,3,5-6	Jack Countryman	Greensburg	812-663-4998	
1,5	Kevin Gold	Indianapolis	317-290-8948	317-543-7098
Maryland				
1,4-7	Ray L. Settle	Arnold	301-647-9192	301-887-0106
1,5,7	Woodrow S. Webster	Fallston	301-879-7034	301-887-0171
Michigan				
1,5,7	Jim Anker	Auburn Hills	313-391-0033	313-544-5344
1,7	Michael McMinn	Swartz Creek	313-655-4442	313-232-6541
1,4-7	Pete Ross	Wayne	313-728-8269	
Minnesota				
1,4-7	James Hirsch	Coon Rapids	612-421-8393	612-422-5572
1	Richard Marchiafava	Fridley	612-572-9305	
Missouri				
1,4-7	Whit Crowley	Manchester	314-394-7955	
Nebraska				
1,7	Kevin Garvin	Dixon	402-584-2271	402-584-2271
5-7	Dr. John W. Kelley	Omaha	402-397-3485	
1,4-7	Larry B. McEwen	Hastings	402-463-2267	402-461-7550
New Hampshire				
1-3	Phil Kirkpatrick	Keene		603-352-0640
New Jersey				
1,4-7	Jay Hubschman	Fairfield	201-575-1968	201-624-8046
1,5,6	David Scott	Wall	201-531-0600	201-531-4016
New Mexico				
1	Gary Young	Corrales	505-897-1770	505-897-1770
New York				
1	Bob Beer	Coram	516-928-6870	
1,5-6	Linda Doscher	West Nyack	914-358-7064	
1,4,7	Carlos M. Madan	Morrisonville	518-562-0779	518-359-3322
7	Larry Merow	Sayville	516-567-0603	516-422-0315
1,4,7	James L. Nicoll	Pittsford	716-381-9480	716-546-6732
1,6,7	Terry Williamson	Orchard Park	716-662-5104	716-873-9750
North Carolina				
2	Marc Apfelstadt	Greensboro	919-282-1494	919-334-5970
Ohio				
3,5,6	Jason Chao	Cleveland Hts.	216-321-5451	216-844-3791
1,4,5	Ltc. Robert Weis	Beavercreek	513-429-4169	513-257-6836
Oregon				
1,5,7	Jim Emig	Portland	503-771-1916	503-280-5666
7	M.W. Fox	Corvallis	503-754-7623	503-737-3628
1,5	Dave Lomax	Lake Oswego	503-636-7289	
Pennsylvania				
4,7	Claude Davis, Jr.	Stewartstown	717-993-6874	717-845-3571
1,5	Martin Friedman	Broomall	215-353-2753	
1,5,7	Richard L. Gable	Pittsburgh	412-963-6158	412-963-1128
7	Joel Perlish	Havertown	215-789-7673	
1,3-7	Bruce Shanker	Warminster	215-674-0118	
1,4-5	Hal Shapiro	Eagleville	215-630-8936	215-922-0500
Texas				
1	Larry Jones	El Paso	915-533-3302	915-565-3016
1,6	Joseph Kline	Lubbock	806-796-0829	
1,5,6	Ramon Merlin	San Antonio	512-496-5331	
Virginia				
1	Peter Pfeiffer	Herndon	703-437-1985	703-834-3618
Canada				
1	Michael Beebe	Victoria	604-477-4630	604-721-7954
1,4,5	Jean Guy Mariage	Montreal	514-844-2932	514-252-2541
1,3	Nick W. Van Helsdingen	Tranquillity Base	604-296-3260	
England				
1,4,7	Andrew C. Letchford	Plymouth	0752766435	44752766435
New Zealand				
1,5	H.P.H. Harrison	Tauranga	075-442-842	075-778-000
Saudi Arabia				
1,7	James E. Hanushek	Dhahran	3-878-4075	3-877-1533

NAUG Membership

Name ______________________________

Member Nº, if renewing ______________________________

Address ______________________________

____

City ____________________ State ____________

Zip ____________ Country ____________________

Home Phone ______________________________

Work Phone ______________________________

The *AppleWorks Forum* — 12 monthly issues, shipped as follows:

Circle One:	One Year	Two[1] Years
2nd Class postage - United States	$29	$58
2nd Class postage - Canada and Mexico	$39	$78
1st Class postage - United States	$44	$88
1st Class airmail - Canada and Mexico	$49	$98
Surface mail outside North America	$41	$82
Airmail outside North America	$64	$128
NAUG on Disk[2]	$84	$168
UltimateWorks - all NAUG items		
Without NAUG on Disk[2]	$99	$198
With NAUG on Disk[2]	$149	$298

Total Enclosed $ ____________

□ Check □ MC/Visa □ PO #[3] ____________

Credit Card Account # ______________________________

Expiration Date ______________________________

Signature ______________________________

[1] Avoids future price increases.
[2] U.S. Price. Foreign orders by credit card only.
[3] Payment must accompany all purchase orders.

NAUG shares members' addresses with other users groups and selected vendors. If you do not want to receive mail from these agencies, check here: □

Classified Ads

FOR SALE: A YEARS SUPPLY OF WISDOM. 1001 clever, clean, famous, funny, witty, though provoking quotations. AppleWorks data base. Easy to use. Be clever. Impress your friends. Use for speeches. Spice up reports. 2 free bonus data bases. Money back guarantee. $5. Quote Disk, Data Base Users Group, 11 Bellflower, Fairport, NY 14450.

INEXPENSIVE CHURCH MANAGEMENT SOFTWARE for Apple II compatible computers using AppleWorks. These application templates include record keeping for membership, finances, contributions, visitation, Sunday school, planning, goal setting, evaluation, worship, and much more. Special data base of music information from hymnals from many denominations. Many other programs. Send for details: Software Sharing Ministries, 2020 North Fremont St., Tacoma, WA 98406; (206) 756-7980.

FAMOUS BIRTHDAYS — Interesting and Fun! AppleWorks Data Base containing 7000+ famous people, their birthdays, what made them famous. Printouts make interesting birthday cards. Specify 3.5 or 5.25, $12.95. Benz Computing, 38 Belton, Stanhope, N.J. 07874; (201) 347-6476.

NATIONWIDE WHOLESALE DIRECTORY. Buy anything at wholesale or below. In today's economic troubles, you can get your buying power back by knowing where to buy at the cheapest prices. 5 1/4" disk only $19.95. Send to HCR, Box 24, Livingston Manor, New York 12758; (914) 439-3536.

MODERN KEYBOARDS. Detached like GS/IBM, low profile for comfort! Great new features for IIe or +: function keys for "CATALOG", "POKE", etc. 12 instantly programmable keys for commands, pathnames, etc. Logical arrow key layout. Owners are delighted! Affordable! Send SASE: Martek Electronics, POB 24, Novi, MI 48050; (313) 348-3812.

NO NEED TO TURN "BLUE" FOR POWER. Custom UltraMacros programming. Steve Beville, 3392 Glenn Springs Rd., Spartanburg, SC 29302. (802) 582-3687.

SOUP UP CLASSIC! is one of the most useful AppleWorks enhancements to appear since UltraMacros, adding power, speed and productivity. 275+ macros! Endorsed by Ruth Witkin, A+/inCider **"AppleWorks In Action"** columnist & author of **"Success With AppleWorks"** and **"The Best Book of AppleWorks"**. Easy to remember memory devices. Macros are organized into default sets, ready to compile and use. SATISFACTION GUARANTEED. $24.95 plus $3 S&H (check or MO). Requires UltraMacros enhanced AppleWorks 2.x or 3.0. Specify AppleWorks version and 3.5" OR 5.25" disks.
Beaumont Software, 5520 Hooks Avenue, Beaumont, Texas 77706; (409) 892-4120.

WANTED: C.A.D. program for Apple IIe or source for the program. Rocky Mtn. Isetta Club c/o V. Cameron, 6516 Constellation Dr., Ft. Collins, Colorado 80525; (303) 226-3544

NAUG members can place classified advertisements in the ***AppleWorks Forum*** for 50¢ per word per issue. Payment must accompany your order which must be received at least 45 days before the cover date of the issue in which the advertisement will appear.

This issue of the ***AppleWorks Forum*** does not include an Electronic Index Update. For a free copy of the January 1991 update, send a self-addressed, stamped envelope to Electronic Index Update, **NAUG**, Box 87453, Canton, Michigan 48187. The complete index is available on disk from **NAUG**'s Public Domain Library on 5.25-inch disks ($6) and 3.5-inch disks ($6), plus $2 s/h. Members can also download the index from **NAUG**'s bulletin board or from the **NAUG** areas on CompuServe and America Online.

Apple-Works Forum

NAUG
National AppleWorks Users Group
Box 87453, Canton, Michigan 48187
(313) 454-1115
BBS (313) 736-8102

TIME VALUE MATERIAL